

## **CONTENT**

Sl. No.	Date	Experiment	Page No.	Signature of Staff in Charge
1	24-03-2022	COMPONENT IDENTIFICATION	4	
2	31-03-2022	LINUX COMMANDS		
3	07-04-2022	LINUX FILE SYSTEM		
4	18-04-2022	SHELL SCRIPT		
5	25-04-2022	LAMP STACK INSTALLATION		
6	05-05-2022	NETWORKING COMMANDS		
7	19-05-2022	VIRTUAL BOX		
8	28-05-2022	WIRESHARK		
9	02-06-2022	LARAVEL SERVER		

## **EXPERIMENT-1**

### **COMPUTER HARDWARE**

#### **Mother Board:-**

A motherboard (also called mainboard, main circuit board, mb, mboard, backplane board, base board, system board, logic board (only in Apple PCs) or mobo) is the main printed circuit board (PCB) in general-purpose computers and other expandable systems. It holds and allows communication between many of the crucial electronic components of a system, such as the central processing unit (CPU) and memory, and provides connectors for other peripherals. Unlike a backplane, a motherboard usually contains significant sub-systems, such as the central processor, the chipset's input/output and memory controllers, interface connectors, and other components integrated for general use.

The motherboard is mounted inside the case and is securely attached via small screws through pre-drilled holes. Motherboard contains ports to connect all of the internal components. It provides a single socket for CPU, whereas for memory, normally one or more slots are available. Motherboards provide ports to attach the floppy drive, hard drive, and optical drives via ribbon cables. Motherboard carries fans and a special port designed for power supply.

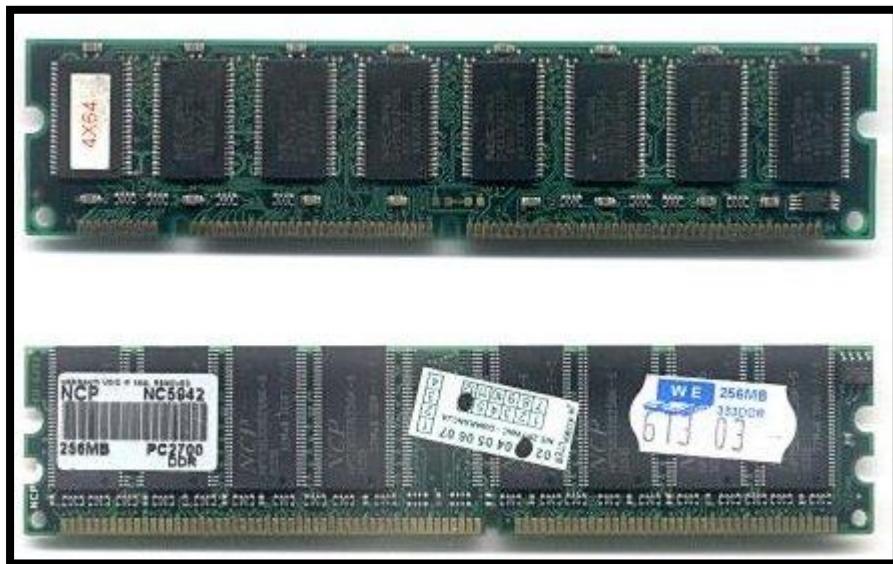
There is a peripheral card slot in front of the motherboard using which video cards, sound cards, and other expansion cards can be connected to the motherboard.

On the left side, motherboards carry a number of ports to connect the monitor, printer, mouse, keyboard, speaker, and network cables. Motherboards also provide USB ports, which allow compatible devices to be connected in plug-in/plug-out fashion. For example, pen drive, digital cameras, etc.



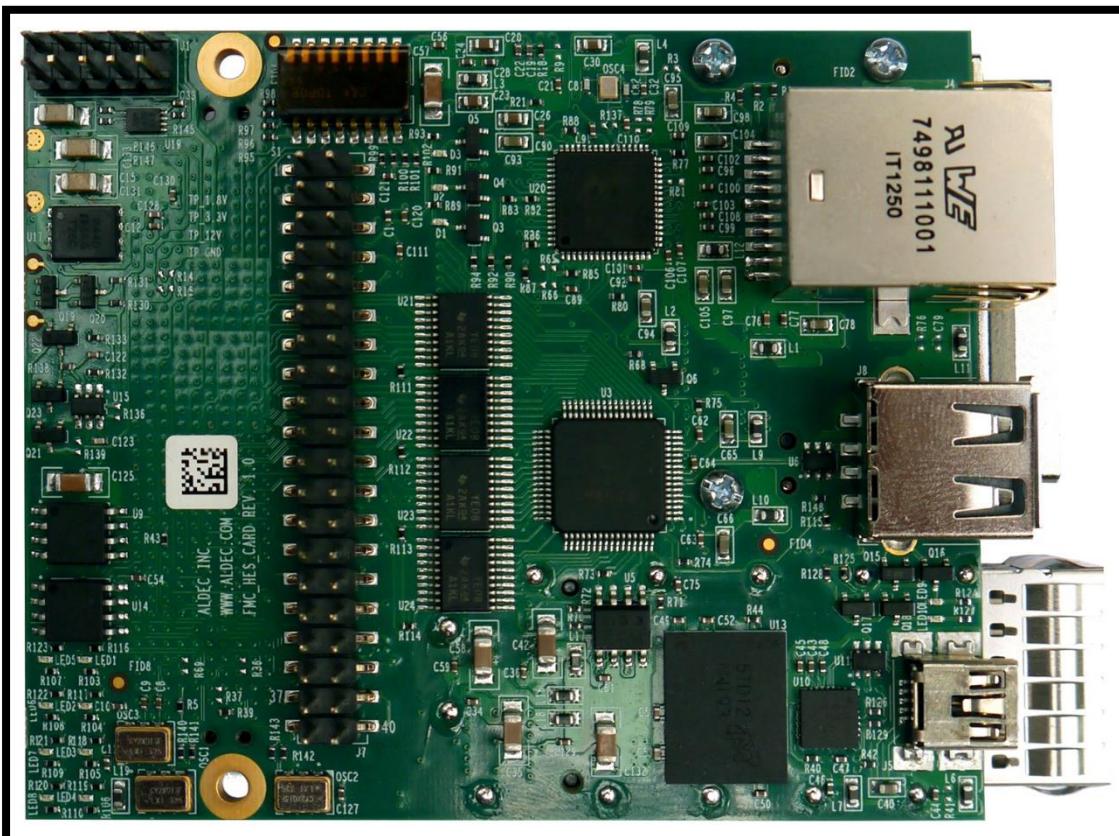
**Ram Modules:-**

In computing, a memory module or RAM (random-access memory) stick is a printed circuit board on which memory integrated circuits are mounted. Memory modules permit easy installation and replacement in electronic systems, especially computers such as personal computers, workstations, and servers. The first memory modules were proprietary designs that were specific to a model of computer from a specific manufacturer.



### **Daughter Cards:-**

A daughterboard is type of circuit board that plugs in or is attached to the motherboard or similar expansion card to extend its features and services. A daughterboard complements the existing functionality of a motherboard or an expansion card. A daughterboard is also known as daughter card, piggyback board, riser card or mezzanine board. A daughterboard is connected directly to the motherboard. Unlike expansion cards, which connect with the motherboard using bus and other serial interfaces, daughterboards are usually directly embedded through soldering. Like a motherboard, a daughterboard has sockets, pins, plugs and connectors to be attached to other boards. Typically, daughterboards are released as a post-launch update to a motherboard or expansion card. For example, a MIDI daughterboard is used to add on the functionality of the sound card.



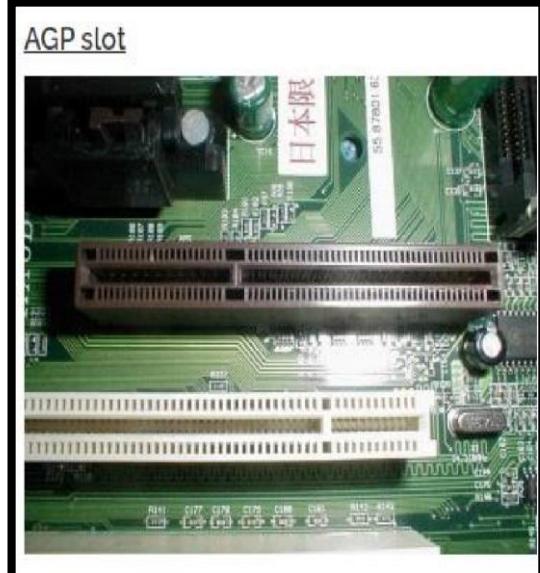
**Bus Slots:-**

An expansion slot refers to any of the slots on a motherboard that can hold an expansion card to expand the computer's functionality, like a video card, network card, or sound card. The expansion card is plugged directly into the expansion port so that the motherboard has direct access to the hardware. However, since all computers have a limited number of expansion slots, it's important to open your computer and check what's available before you buy one. Some older systems require the use of a riser board to add additional expansion cards; however, modern computers not only usually have enough expansion slot options, but they also have features integrated directly into the motherboard, eliminating the need for so many expansion cards. There are three different types of expansion slots: PCI Express, PCI, and AGP.

**PCI (Peripheral Component Interconnect) Slot :** The PCI slot is the most common form of internal expansion for a PC. Some PCs have a mixture of PCI and PCI Express slots.

**PCI express (PCIe) Slots :** The best type of expansion slot to have in your PC is the PCI Express. The PCI Express type of expansion slot communicates with the motherboard, and therefore with the microprocessor, both quickly and efficiently.

**AGP (Accelerated Graphics Port) Slot :** This type of expansion slot was specifically designed to deal with graphics adapters. In fact, AGP stands for Accelerated Graphics Port. Older PCs may sport this expansion slot, but the best video cards use PCI Express.



**SMPS:-**

A switched-mode power supply (SMPS) is an electronic circuit that converts power using switching devices that are turned on and off at high frequencies, and storage components such as inductors or capacitors to supply power when the switching device is in its non-conduction state.

Switching power supplies have high efficiency and are widely used in a variety of electronic equipment, including computers and other sensitive equipment requiring stable and efficient power supply.

A switched-mode power supply is also known as a switch-mode power supply or switching-mode power supply.

Switched-mode power supplies are classified according to the type of input and output voltages. The four major categories are:

- AC to DC
- DC to DC
- DC to AC
- AC to AC



**Internal Storage Devices:-**

Some storage devices are classed as 'internal' which means they are inside the computer case. Most computers have some form of internal storage. The most common type of internal storage is the hard disk. At the most basic level, internal storage is needed to hold the operating system so that the computer is able to access the input and output devices. It will also be used to store the applications software that you use and more than likely, the original copies of your data files. Internal storage allows the data and applications to be loaded very rapidly into memory, ready for use. The data can be accessed much faster than data which is stored on an external storage device. This is because internal storage devices are connected directly to the motherboard and its data bus whereas external devices are connected through a hardware interface such as USB, which means they are considerably slower to access. Internal storage also means that if the computer is moved around, it will still retain its most commonly used data. The main disadvantage of internal storage is that when the hard disk fails (and it will), all the data and applications may be lost. This can be avoided to some extent by using more than one hard disk within the machine. Each hard disk has a copy of all the data, so if one fails the other can carry on. This is called a RAID array. An alternative is to use external drives for backup.



**Interfacing Ports:-**

A Computer Port is an interface or a point of connection between the computer and its peripheral devices. Some of the common peripherals are mouse, keyboard, monitor or display unit, printer, speaker, flash drive etc. The main function of a computer port is to act as a point of attachment, where the cable from the peripheral can be plugged in and allows data to flow from and to the device.

**Types of ports:**

**Serial Port** - used for external modems and older computer mouse.

**Parallel Port** - used for scanners and printers.

**PS/2 Port**- used for old computer keyboard and mouse.

**Universal Serial Bus (or USB) Port** - It can connect all kinds of external USB devices such as external hard disk, printer, scanner, mouse, keyboard, etc.

**VGA Port** -connects monitor to a computer's video card. It has 15 holes. Similar to the serial port connector. However, serial port connector has pins, VGA port has holes.

**Power Connector** -connects to the computer's power cable that plugs into a power bar or wall socket.

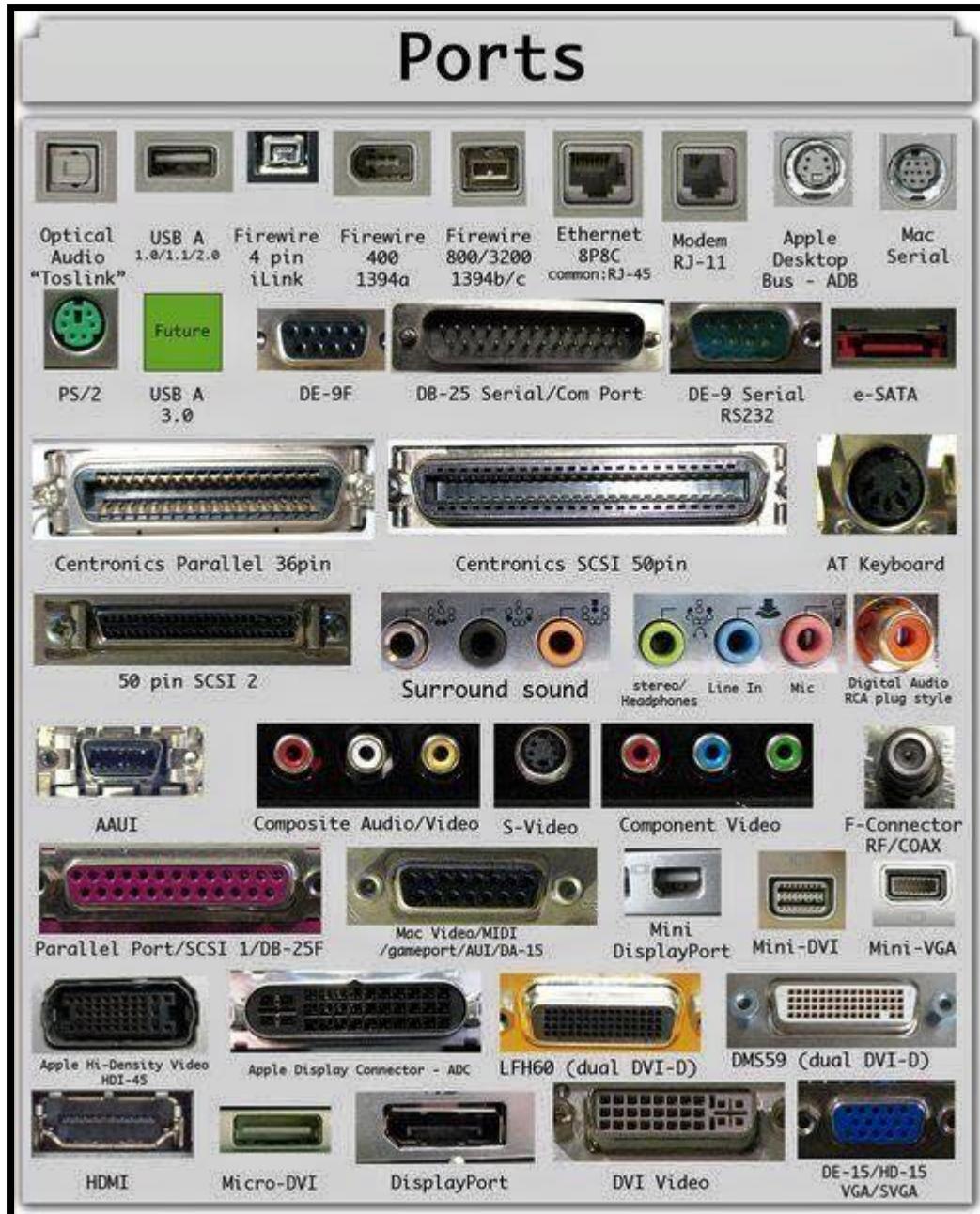
**Modem Port** - connects a PC's modem to the telephone network.

**Ethernet Port** - connects to a network and high speed Internet. Connects the network cable to a computer.

**Game Port** - connect a joystick to a PC. Now replaced by USB Digital Video Interface

**DVI port** - connects Flat panel LCD monitor to the computer's high-end video graphic cards.

**Sockets** - sockets connect the microphone and speakers to the sound card of the computer.



## EXPERIMENT-2

### LINUX COMMANDS

#### Man

```
LS(1)                               User Commands                   LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILEs (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
    Manual page ls(1) line 1 (press h for help or q to quit)
```

#### Ls

```
stud@debian:~$ ls
24_ANGELDBMS  Documents  networks
48            Downloads  noel45
48A           gaya3     NSALAB_1.SH
abel          hisham63  Pictures
ADS_45         home      Public
akhila16      Home      public_html
anagha        india     python_45
anagha02      jane      python_45.py
Anusree37     java_45   python_46
ayana          LAB       python_54
b              MASM     shadasm45.java
bivina         mca      Templates
de45          minwa    test
de45.java     Music    Videos
Desktop       n        'VirtualBox VMs'
dev           network_45

stud@debian:~$
```

**Echo**

```
user@user:~/network$ echo 'network'
network
user@user:~/network$ █
```

**Read**

```
user@user:~/network$ echo 'enter your name'
enter your name
user@user:~/network$ read name
user
user@user:~/network$ echo $name
user
█
```

**Cat**

```
stud@debian:~/networks$ cat >file1
Computer networks
stud@debian:~/networks$ cat file1
Computer networks
stud@debian:~/networks$ █
```

**More**

A screenshot of a terminal window titled "stud@debian: ~". The window contains the following text:

```
stud@debian:~$ more file3
kerala
tamilnadu
stud@debian:~$ █
```

**Less**

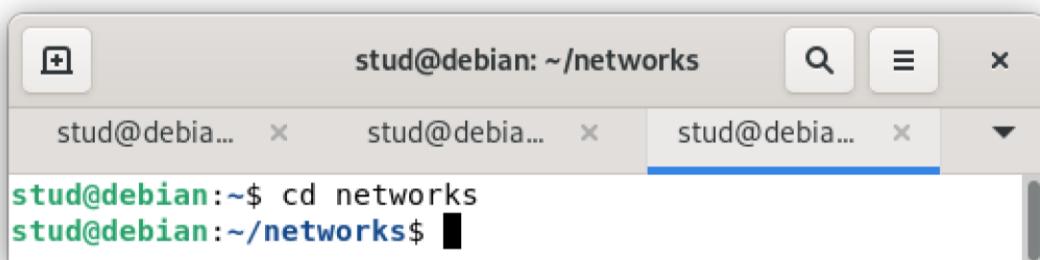
A screenshot of a terminal window titled "stud@debian: ~". The window contains the following text:

```
stud@debian:~$ less file3
kerala
tamilnadu
stud@debian:~$ █
```

### Cat

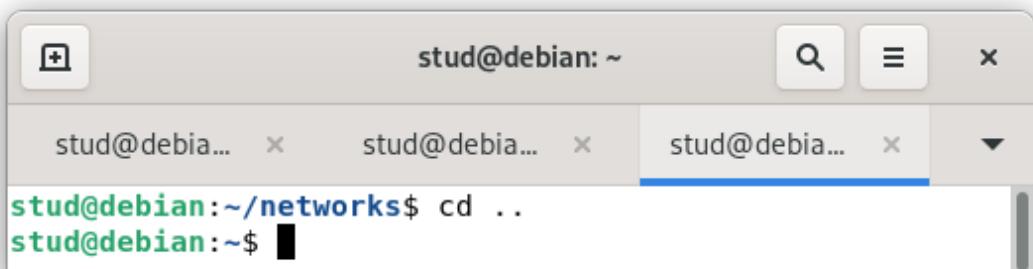
```
stud@debian:~/networks$ cat >file1  
Computer networks  
stud@debian:~/networks$ cat file1  
Computer networks  
stud@debian:~/networks$
```

### Cd



A screenshot of a terminal window titled "stud@debian: ~/networks". The window has three tabs at the top: "stud@debia...", "stud@debia...", and "stud@debia...". The third tab is active. The terminal shows the command "cd networks" being entered and executed, followed by the prompt "stud@debian:~/networks\$".

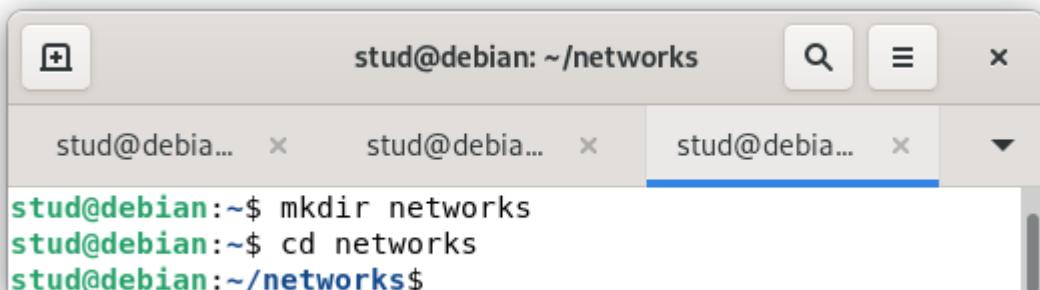
```
stud@debian:~$ cd networks  
stud@debian:~/networks$
```



A screenshot of a terminal window titled "stud@debian: ~". The window has three tabs at the top: "stud@debia...", "stud@debia...", and "stud@debia...". The third tab is active. The terminal shows the command "cd .." being entered and executed, followed by the prompt "stud@debian:~\$".

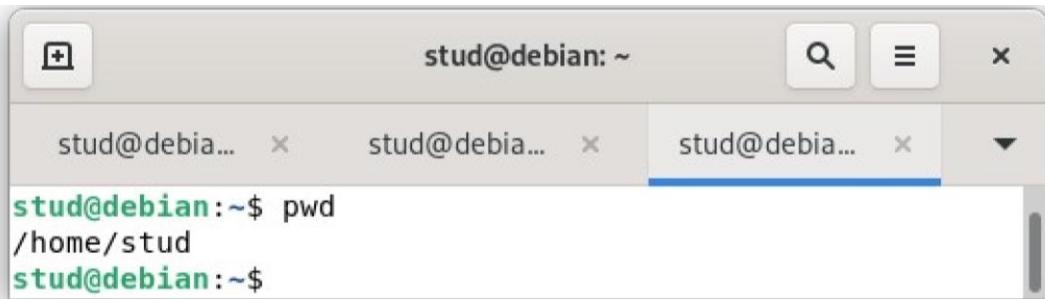
```
stud@debian:~/networks$ cd ..  
stud@debian:~$
```

### Mkdir



A screenshot of a terminal window titled "stud@debian: ~/networks". The window has three tabs at the top: "stud@debia...", "stud@debia...", and "stud@debia...". The third tab is active. The terminal shows the commands "mkdir networks" and "cd networks" being entered and executed, followed by the prompt "stud@debian:~/networks\$".

```
stud@debian:~$ mkdir networks  
stud@debian:~$ cd networks  
stud@debian:~/networks$
```

**Pwd**

A screenshot of a terminal window titled "stud@debian: ~". The window has three tabs at the top: "stud@debia...", "stud@debia...", and "stud@debia...". The third tab is active. The terminal output shows:

```
stud@debian:~$ pwd
/home/stud
stud@debian:~$
```

**Find**

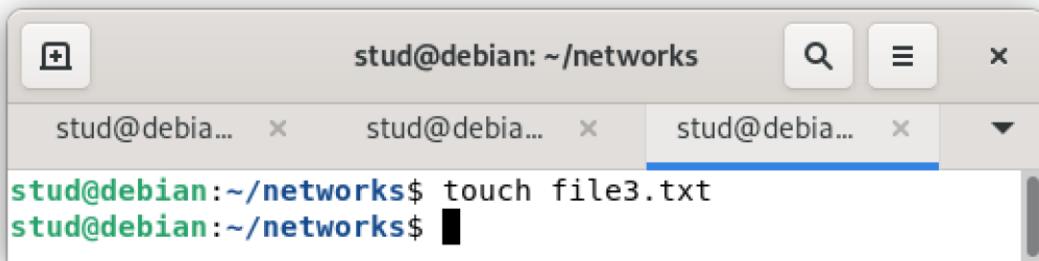
```
stud@debian:~$ cd ACN
stud@debian:~/ACN$ touch f1 f2
stud@debian:~/ACN$ cd ..
stud@debian:~$
stud@debian:~$ find ./ACN -name f1
./ACN/f1
```

**Mv**

```
stud@debian:~$ mkdir ACN
stud@debian:~$ touch f1.txt
stud@debian:~$ touch f2.txt
stud@debian:~$ mv f1.txt ./ACN
stud@debian:~$ cd ACN
stud@debian:~/ACN$ ls
f1.txt
stud@debian:~/ACN$ █
```

**Cp**

```
stud@debian:~/networks$ cp file4 file3
stud@debian:~/networks$ cat file3
1
2
3
4
5
6
7
8
9
10
stud@debian:~/networks$ █
```

**Rm**


stud@debian: ~/networks

```
stud@debian:~/networks$ touch file3.txt
stud@debian:~/networks$
```

**Tar**

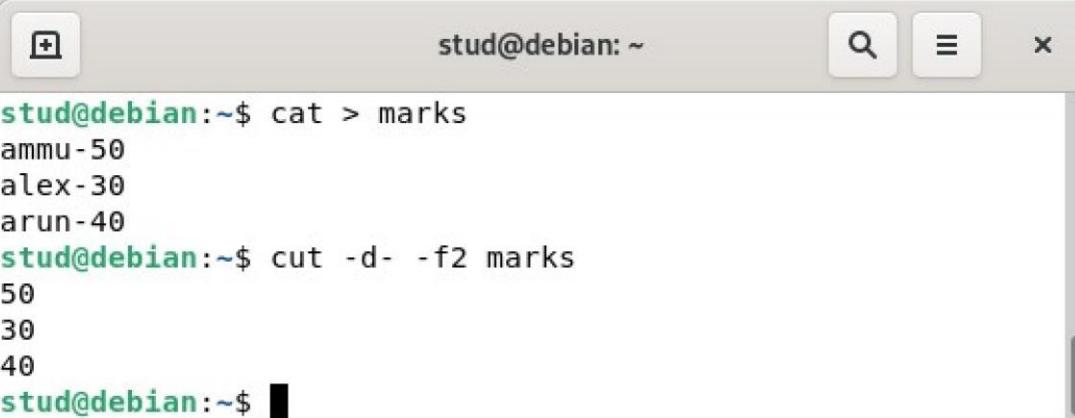
```
stud@debian:~$ touch file{0..100}.txt
stud@debian:~$ tar cf file.tar file
stud@debian:~$ ls -l
total 348
drwxr-xr-x 2 stud stud 4096 Jun  9 10:48 1234
-rw-r--r-- 1 stud stud    79 Jun 16 15:00 6.sh
-rw-r--r-- 1 stud stud   18 Jun 14 10:47 a
-rw-r--r-- 1 stud stud    47 Jan  5 11:53 afiedt.buf
drwxr-xr-x 2 stud stud 4096 Jun 16 15:15 Anagha
drwxr-xr-x 2 stud stud 4096 Apr 29 16:11 ANANDAKRISHNAN.M.S
drwxr-xr-x 2 stud stud 4096 Dec 17 16:26 aparna
drwxr-xr-x 2 stud stud 4096 Jun 14 15:32 archana
drwxr-xr-x 2 stud stud 4096 Nov 17 2021 Arshia42
drwxr-xr-x 6 stud stud 4096 Mar  3 09:40 ATHIRA42
drwxr-xr-x 2 stud stud 4096 May 24 13:31 azad13
-rw-r--r-- 1 stud stud   367 Mar  2 14:13 bill.php
-rwrxr-xr-x 1 stud stud 16760 Jun 14 10:47 c
-rw-r--r-- 1 stud stud    303 Jun 14 10:45 catfile.c
-rw-r--r-- 1 stud stud 1598 Sep 23 2021 CurrentThreadDemo.class
-rw-r--r-- 1 stud stud    526 Sep 23 2021 CurrentThreadDemo.java
drwxr-xr-x 2 stud stud 4096 May 21 11:03 Desktop
drwxr-xr-x 3 stud stud 4096 Sep 28 2021 dev
drwxr-xr-x 2 stud stud 4096 May 21 11:30 Documents
drwxr-xr-x 3 stud stud 4096 Jun 16 15:13 Downloads
-rw-r--r-- 1 stud stud    19 Jun 14 10:39 f1.txt
-rw-r--r-- 1 stud stud    60 Jun 16 15:10 f2
-rw-r--r-- 1 stud stud   714 Jun 14 12:37 fcfs.c
-rw-r--r-- 1 stud stud   212 Sep 28 2021 fibo.sql
drwxr-xr-x 2 stud stud 4096 Jun 16 15:17 file
-rw-r--r-- 1 stud stud     0 Jun 16 15:17 file0.txt
-rw-r--r-- 1 stud stud     0 Jun 16 15:17 file100.txt
-rw-r--r-- 1 stud stud     0 Jun 16 15:17 file10.txt
-rw-r--r-- 1 stud stud     0 Jun 16 15:17 file11.txt
-rw-r--r-- 1 stud stud     0 Jun 16 15:17 file12.txt
```

Wc



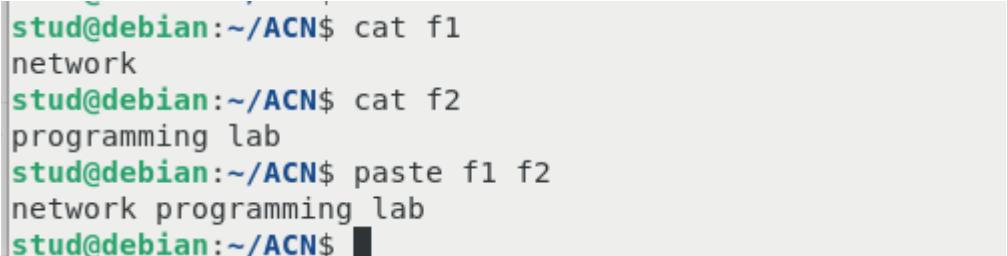
```
stud@debian:~$ wc file3
2 2 17 file3
stud@debian:~$
```

Cut



```
stud@debian:~$ cat > marks
ammu-50
alex-30
arun-40
stud@debian:~$ cut -d- -f2 marks
50
30
40
stud@debian:~$
```

Paste

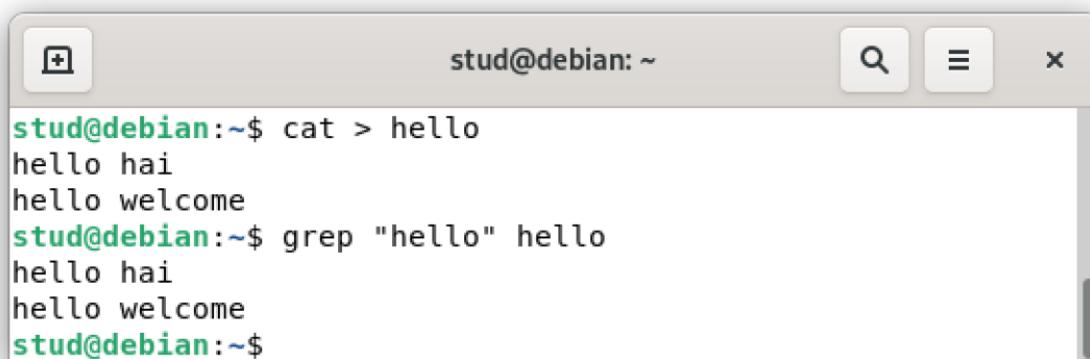


```
stud@debian:~/ACN$ cat f1
network
stud@debian:~/ACN$ cat f2
programming lab
stud@debian:~/ACN$ paste f1 f2
network programming lab
stud@debian:~/ACN$
```

## Head and Tail

```
stud@debian:~/networks$ cat > file4
1
2
3
4
5
6
7
8
9
10
stud@debian:~/networks$ head -8 file4|tail -2
7
8
stud@debian:~/networks$
```

## Grep



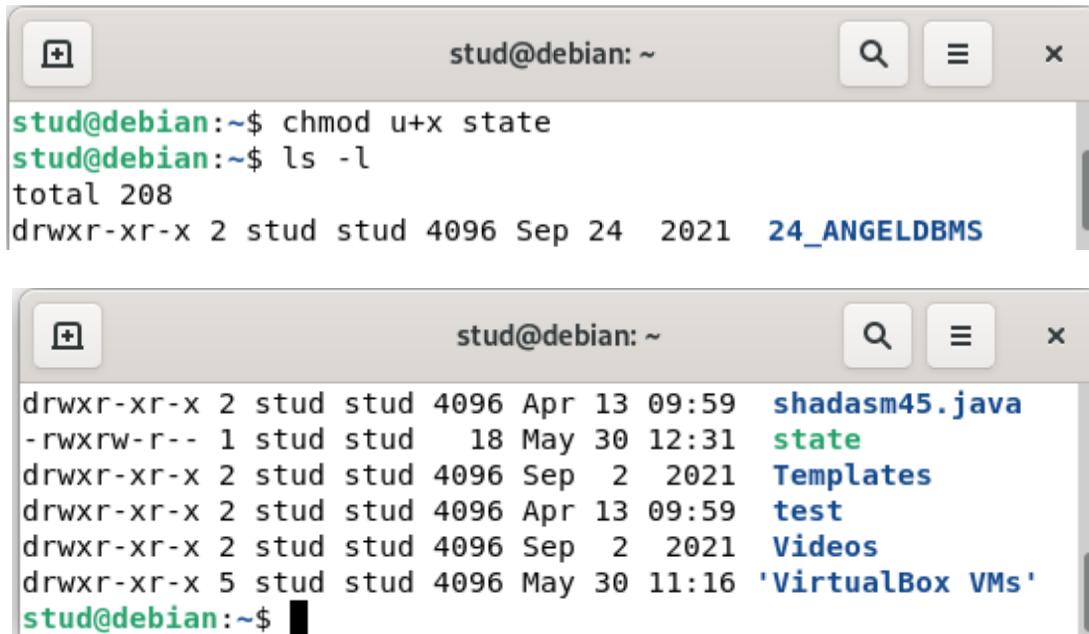
A screenshot of a terminal window titled "stud@debian: ~". The window contains the following text:

```
stud@debian:~$ cat > hello
hello hai
hello welcome
stud@debian:~$ grep "hello" hello
hello hai
hello welcome
stud@debian:~$
```

## Expr

```
user@user:~/network$ expr 4 + 5
9
user@user:~/network$ expr 10 - 5
5
user@user:~/network$ touch file1
```

## Chmod



```
stud@debian:~$ chmod u+x state
stud@debian:~$ ls -l
total 208
drwxr-xr-x 2 stud stud 4096 Sep 24 2021 24_ANGELDBMS
```

```
stud@debian:~$ ls -l
drwxr-xr-x 2 stud stud 4096 Apr 13 09:59 shadasm45.java
-rwxrw-r-- 1 stud stud 18 May 30 12:31 state
drwxr-xr-x 2 stud stud 4096 Sep 2 2021 Templates
drwxr-xr-x 2 stud stud 4096 Apr 13 09:59 test
drwxr-xr-x 2 stud stud 4096 Sep 2 2021 Videos
drwxr-xr-x 5 stud stud 4096 May 30 11:16 'VirtualBox VMs'
stud@debian:~$
```

## Redirections

```
user@user:~/network$ ls -l >> q8.sh
user@user:~/network$ cat q8.sh
total 16
-rwxrw-r-- 1 user user 61 Jun 9 21:39 f6.sh
-rw-rw-r-- 1 user user 79 Jun 10 06:04 f7.sh
-rwxrw-r-- 1 user user 248 Jun 16 11:55 q13.sh
-rw-rw-r-- 1 user user 104 Jun 16 06:59 q4.sh
-rw-rw-r-- 1 user user 0 Jun 16 12:20 q8.sh
```

## Pipes

```
stud@debian:~$ ls *.sh | cat >f2
stud@debian:~$ cat f2
6.sh
file1.sh
if.sh
leapyear.sh
sintst.sh
string.sh
teat.sh
stud@debian:~$
```

## **EXPERIMENT-3**

### **LINUX FILE SYSTEM**

It makes sense to explore the Linux file system from a terminal window. In fact, that is the name of the first tool you'll install to help you on the way: tree. If you are using Ubuntu or Debian, you can do:

```
sudo apt install tree
```

Once installed, stay in your terminal window and run tree like this :

```
$tree/
```

The / in the instruction above refers to the root directory. The root directory is the one from which all other directories branch off from. When you run tree and tell it to start with /, you will see the whole directory tree, all directories and all the subdirectories in the whole system, with all their files, flyby.

If you have been using your system for some time, this may take a while, because, even if you haven't generated many files yourself, a Linux system and its apps are always logging, caching, and storing temporary files. The number of entries in the file system can grow quite quickly.

Instead, try this :

**tree-L1/**

```
(base) sunil@debian:~$ tree -L 1 /
/
├── bin    -> usr/bin
├── boot
├── dev
├── etc
├── home
├── initrd.img    -> boot/initrd.img-4.19.0-14-amd64
├── initrd.img.old    -> boot/initrd.img-4.19.0-13-amd64
├── lib    -> usr/lib
├── lib32   -> usr/lib32
├── lib64   -> usr/lib64
├── libx32  -> usr/libx32
├── lost+found
├── media
├── mnt
├── opt
├── proc
├── root
├── run
├── sbin   -> usr/sbin
├── snap
├── srv
├── sys
└── tmp
/usr
/var
vmlinuz    -> boot/vmlinuz-4.19.0-14-amd64
vmlinuz.old    -> boot/vmlinuz-4.19.0-13-amd64

23 directories, 4 files
(base) sunil@debian:~$
```

And you should see a listing similar to what is shown in the Figure

The instruction above can be translated as “show me only the 1st Level of the directory tree starting at /(root)“.The –L option tells tree how many levels down you want to see.

Most Linux distributions will show you the same or a very similar layout to what you can see in the image above. This means that even if you feel confused now, master this, and you will have a handle on most, if not all, Linux installations in the whole wide world.

Now, let's look at what each directory is used for. While we go through each, you can peek at their contents using ls.

## Directories

From top to bottom, the directories you are seeing are as follows.

### **/bin**

/bin is the directory that contains binaries, that is, some of the applications and programs you can run. You will find the ls program mentioned above in this directory, as well as other basic tools for making and removing files and directories, moving them around, and so on. There are more bin directories in other parts of the file system tree, but we'll be talking about those in a minute.

### **/boot**

The /boot directory contains files required for starting your system. If you mess up one of the files in here, you may not be able to run your Linux and it is a pain to repair. On the otherhand, don't worry too much about destroying your system by accident: you have to have super user privileges to do that.

### **/dev**

/dev contains device files. Many of these are generated at boot time or even on the fly. For example, if you plug in a new webcam or a USB pendrive into your machine, a new device entry will automatically pop up here.

### **/etc**

/etc is the directory where names start to get confusing. /etc gets its name from the earliest Unixes and it was literally “et cetera” because it was the dumping ground for system files administrators were not sure where else to put.

Nowadays, it would be more appropriate to say that etc stands for “Everything to configure”, as it contains most, if not all system-wide configuration files. For example, the files that contain the name of your system, the users and their passwords, the names of machines on your network and when and where the partitions on your hard disks should be mounted are all in here. Again, if you are new to Linux, it may be best if you don't touch too much in here until you have a better understanding of how things work.

### **/home**

/home is where you will find your users' personal directories. In my case, under /home there are two directories: /home/paul, which contains all my stuff; and /home/guest, in case anybody needs to borrow my computer.

### **/lib**

/lib is where libraries live. Libraries are files containing code that your applications can use. They contain snippets of code that applications use to draw windows on your desktop, control

peripherals, or send files to your hard disk.

There are more lib directories scattered around the file system, but this one, the one hanging directly off of / is special in that, among other things, it contains the all-important kernel

modules. The kernel modules are drivers that make things like your video card, sound card, WiFi, printer, and so on, work.

### **/media**

The /media directory is where external storage will be automatically mounted when you plug it in and try to access it. As opposed to most of the other items on this list, /media does not hail back to 1970s, mainly because inserting and detecting storage (pendrives, USB hard disks, SD cards, external SSDs, etc) on the fly, while a computer is running, is a relatively new thing.

### **/mnt**

The /mnt directory, however, is a bit of remnant from days gone by. This is where you would manually mount storage devices or partitions. It is not used very often nowadays.

### **/opt**

The /opt directory is often where software you compile (that is, you build yourself from source code and do not install from your distribution repositories) sometimes lands.

Applications will end up in the /opt/bin directory and libraries in the /opt/lib directory.

A slight digression: Another place where applications and libraries end up in is /usr/local. When software gets installed here, there will also be /usr/local/bin and /usr/local/lib directories. What determines which software goes where is how the developers have configured the files that control the compilation and installation process

### **/proc**

/proc, like /dev is a virtual directory. It contains information about your computer, such as information about your CPU and the kernel your Linux system is running. As with /dev, the files and directories are generated when your computer starts, or on the fly, as your system is running and things change.

### **/root**

/root is the home directory of the super user(also known as the “Administrator”) of the system. It is separate from the rest of the users’ home directories because you are not meant to touch it. Keep your own stuff in your own directories, people.

### **/run**

/run is another new directory. System processes use it to store temporary data for their own nefarious reasons.

**/sbin**

/sbin is similar to /bin, but it contains applications that only the super user (hence the initials) will need. You can use these applications with the sudo command that temporarily concedes you super user powers on many distributions. /sbin typically contains tools that can install stuff, delete stuff and format stuff. As you can imagine, some of these instructions are lethal if you use them improperly, so handle with care.

**/usr**

The /usr directory was where users' home directories were originally kept back in the early days of UNIX. However, now/home is where users kept their stuff as we saw above. These days,/usr contains a mish- mash of directories which in turn contain applications, libraries, documentation, wallpapers, icons and a long list of other stuff that need to be shared by applications and services.

You will also find bin, sbin and lib directories in /usr. What is the difference with their root hanging cousins? Not much nowadays. Originally, the/bin directory (hanging off of root)would contain very basic commands, like ls, mv and rm; the kind of commands that would come pre-installed in all UNIX/Linux installations, the bare minimum to run and maintain a system. /usr/bin on the other hand would contain stuff the users would install and run to use the system as a work station, things like word processors, web browsers, and other apps.

But many modern Linux distributions just put everything into /usr/bin and have/bin point to /usr/bin just in case rasing it completely would break something. So, while Debian, Ubuntu and Mint still keep /bin and /usr/bin (and /sbin and /usr/sbin) separate; others, like Arch and its derivatives just have one “real” directory for binaries, /usr/bin, and the rest or \*bins are “fake” directories that point to /usr/bin.

**/srv**

The/srv directory contains data for servers. If you are running a web server from your Linuxbox, your HTML files for your sites would go into /srv/http (or /srv/www). If you were running an FTP server, your files would go into /srv/ftp.

**/sys**

/sys is another virtual directory like /proc and /dev and also contains information from devices connected to your computer.

In some cases you can also manipulate those devices. I can, for example, change the brightness of the screen of my laptop by modifying the value stored in the/sys/devices/pci0000:00/0000:00:02.0/drm/card1/card1-eDP-1/intel\_backlight/brightness file(on your machine you will probably have a different file). But to do that you have to

become super user. The reason for that is, as with so many other virtual directories, messing

with the contents and files in /sys can be dangerous and you can trash your system. DO NOT TOUCH until you are sure you know what you are doing.

### **/tmp**

/tmp contains temporary files, usually placed there by applications that you are running. The files and directories often (not always) contain data that an application doesn't need right now, but may need later on.

You can also use /tmp to store your own temporary files — /tmp is one of the few directories hanging off / that you can actually interact with without becoming super user.

### **/var**

/var was originally given its name because its contents was deemed variable, in that it changed frequently. Today it is a bit of a misnomer because there are many other directories that also contain data that changes frequently, especially the virtual directories we saw above. Be that as it may, /var contains things like logs in the /var/log subdirectories. Logs are files that register events that happen on the system. If something fails in the kernel, it will be logged in a file in /var/log; if someone tries to break into your computer from outside, your firewall will also log the attempt here. It also contains spools for tasks. These “tasks” can be the jobs you send to a shared printer when you have to wait because another user is printing along document, or mail that is waiting to be delivered to users on the system.

Your system may have some more directories we haven't mentioned above. In the screenshot, for example, there is a /snap directory. That's because the shot was captured on an Ubuntu system. Ubuntu has recently incorporated snap packages as a way of distributing software.

The /snap directory contains all the files and the software installed from snaps.

## **EXPERIMENT-4**

### **SHELL SCRIPT**

1. Write a shell script to find the sum, the average and the product of the four integers entered.

#### **Shell Script**

```
echo "Enter your first number"  
read a  
  
echo "Enter your second number"  
read b  
  
echo "Enter your third number"  
read c  
  
echo "Enter your fourth number"  
read d  
  
sum=$(($a + $b +$c + $d))  
  
avg=$(($sum / 4))  
  
prod=$(($a * $b * $c * $d))  
  
echo "The sum of these numbers is: "$sum  
  
echo "The average of these numbers is: "$avg  
  
echo "The product of these numbers is: "$prod
```

### Output

```
stud@debian:~$ vi NSA_1.SH
stud@debian:~$ bash NSA_1.SH
Please enter your first number:
1
Second number:
2
Third number:
3
Fourth number:
4
The sum of these number is:10
The average of these numbers is:2.5000000000000000000000000000000
The product of these number is:24
stud@debian:~$ █
```

2 .a) Write a program to check whether a number entered is odd or even.

### Shell Script

```
echo "Enter a number"  
read n  
echo "Result is"  
if [ `expr $n % 2` -eq 0 ]  
then  
    echo "$n is even"  
else  
    echo "$n is odd"  
fi
```

### Output

```
stud@debian:~$ vi 2.sh  
stud@debian:~$ bash 2.sh  
Enter a number:  
5  
RESULT:  
5 is Odd  
stud@debian:~$ □
```

b) Write a shell script to print given number in reverse order.

### **Shell Script**

```
echo "Enter a number:"  
  
read n  
  
sd=0  
  
rev=0  
  
while [ $n -gt 0 ]  
  
do  
  
    sd=$(( $n % 10 ))  
  
    rev=$(( $rev * 10 + $sd ))  
  
    n=$(( $n / 10 ))  
  
done  
  
echo "Reverse number of entered digit is $rev"
```

### **Output**

```
stud@debian:~$ vi 3.sh  
stud@debian:~$ bash 3.sh  
Enter a number :  
94  
Reverse number of entered digit is 49  
stud@debian:~$ █
```

c) Write a shell script to print sum of all digits of a given number.

### Shell Script

```
echo "Enter a number"

read num

sum=0

while [ $num -gt 0 ]

do

    mod=$((num % 10))

    sum=$((sum + mod))

    num=$((num / 10))

done

echo $sum
```

### Output

```
stud@debian:~$ vi 4.sh
stud@debian:~$ bash 4.sh
Enter a number
205
7
stud@debian:~$ █
```

3. Write a shell script that accepts any year from the keyboard and determine whether the year is a leap year or not.

### **Shell Script**

```
echo "Enter a year"  
read year  
  
if [ `expr $year % 400` == 0 ] && [ `expr $year % 100` != 0 ] || [ `expr $year % 4` == 0 ]  
then  
    echo "It is a leap year"  
  
else  
    echo "It is not a leap year"  
fi
```

### **Output**

```
stud@debian:~$ vi 3a.sh  
stud@debian:~$ bash 3a.sh  
2000  
2000 is a leap year  
stud@debian:~$ bash 3a.sh  
1984  
1984 is a leap year  
stud@debian:~$ bash 3a.sh  
3025  
3025 is not a leap year  
stud@debian:~$ █
```

4. Write a shell script to find the factorial of a number.

### Shell Script

```
echo "Enter a number"

read num

temp=$num

fact=1

while [ $num -ge 1 ]

do

    fact=$((fact * $num))

    num=$((num-1))

done

echo "The factorial of $temp is $fact"
```

### Output

```
stud@debian:~$ vi 3B.sh
stud@debian:~$ bash 3B.sh
Enter a number
3
Factorial is 6
stud@debian:~$ █
```

5. Write a program to print prime numbers from 1 to 100.

### Shell Script

```
echo "Prime numbers from 1 to 100 are"
```

```
echo 2
```

```
j=3
```

```
while [ $j -le 100 ]
```

```
do
```

```
i=2
```

```
a=`expr $j - 1`
```

```
while [ $i -le $a ]
```

```
do
```

```
if [ `expr $j % $i` -ne 0 ]
```

```
then
```

```
i=`expr $i + 1`
```

```
else
```

```
break
```

```
fi
```

```
done
```

```
if [ $i -eq $j ]
```

```
then
```

```
echo $j
```

```
fi
```

```
j=`expr $j + 1`
```

done

### Output

```
stud@debian:~$ vi 5.sh
stud@debian:~$ bash 5.sh
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
stud@debian:~$ █
```

6. Write a script for Printing Numbers in ascending Order.

### **Shell Script**

```
echo "Before Sorting"  
  
cat numbers.txt  
  
echo "After Sorting"  
  
sort -n numbers.txt
```

### **Output**

```
stud@debian:~$ bash 6.sh  
Before Sorting  
20  
12  
15  
13  
23  
17  
After Sorting  
12  
13  
15  
17  
20  
23
```

7. Write a shell script which displays a list of files in the current directory to which we have read write and execute permissions.

### **Shell Script**

```
echo "enter the directory name"

read dir if [ -d $dir ] then cd $dir

ls > f exec < f while read line do

if [ -f $line ] then

if [ -r $line -a -w $line -a -x $line ] then

echo "$line has all permissions" else

echo "files not having all permissions" fi fi

done

fi
```

### **Output**

```
stud@debian:~$ vi directory.sh
stud@debian:~$ bash directory.sh
enter the directory name
emelsha55
files not having all permissions
stud@debian:~$ bash directory.sh
enter the directory name
jat
files not having all permissions
stud@debian:~$ █
```

8. Write a shell script to find the number of occurrence of particular digit in an inputted number.

### **Shell Script**

```
count=0

echo "Enter a number "

read num

x=$num

echo "Enter the digit whose occurrence has to be found "

read digit

while [ $num -ne 0 ]

do

    rem=`expr $num % 10`


    if [ $rem -eq $digit ]

        then

            count=`expr $count + 1`


        fi


    num=`expr $num / 10`


done

echo "Number of occurrences of digit \"$digit \"is \"$count\" times"
```

### Output

```
stud@debian:~$ vi occurance.sh
stud@debian:~$ bash occurance.sh
Enter a number
235
Enter the digit whose occurrence has to be found
3
Number of occurrences of digit 3 is 1 times
stud@debian:~$ bash occurance.sh
Enter a number
55570
Enter the digit whose occurrence has to be found
5
Number of occurrences of digit 5 is 3 times
stud@debian:~$ █
```

9. Write shell script to print alternate digits when a 7 digit number is passed as input.

### **Shell Script**

```
echo "Enter a seven digit number"

read num

n=1

while [ $n -le 7 ]

do

    a=`echo $num | cut -c $n`

    echo $a

    n=`expr $n + 2`


done
```

### **Output**

```
stud@debian:~$ vi q9.sh
stud@debian:~$ bash q9.sh
Enter a 7 digit number
7419813
7
1
8
3
stud@debian:~$ █
```

10. Write a shell script that takes a command –line argument and reports on whether it is directory, a file, or something else.

**Shell Script**

```
echo Enter file
read str
if test -f $str
then
    echo File exists and it is an ordinary File
elif test -d $str
then
    echo It is a Directory File
else
    echo File does not exists
fi
if test -c $str
then
    echo It is a Character Device File
fi
```

### Output

```
stud@debian:~$ vi q10.sh
stud@debian:~$ bash q10.sh
Enter file
q9.png
File exists and it is an ordinary file
stud@debian:~$ bash q10.sh
Enter file
yth
Not exists
stud@debian:~$ █
```

11. Write a shell script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers.

### **Shell Script**

```
echo "Enter the Filename"  
read fname  
  
echo "Enter the starting line number"  
read s  
  
echo "Enter the ending line number"  
read n  
  
sed -n $s,$n\p $fname | cat > newline  
  
cat newline
```

### **Output**

```
stud@debian:~$ vi q11.sh  
stud@debian:~$ bash q11.sh  
enter the filename  
directory.sh  
enter the starting line number  
2  
enter the ending line number  
6  
read dir  
if [ -d $dir ]  
then  
cd $dir  
ls > f
```

12. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.

### **Shell Script**

```
#!/bin/bash

if [ $# -lt 2 ]

then

    echo "Usage: ./02dellines1.sh <search_word><file_name> [<file_name> ...]"

exit 1

fi

search_word=$1

for file_name in "$@"

do

    if [ "$file_name" = "$1" ]

    then

        continue

    fi

    echo $file_name

    echo "-----"

    if [ ! -f $file_name ]

    then

        echo "File \"$file_name\" does not exist"

    exit 2

    fi

    while read line
```

```
do  
    if ( !( echo $line | grep "$search_word">>/dev/null ) )  
    then  
        echo $line  
    fi  
done < $file_name  
done
```

### Output

```
stud@debian:~/48A$ vi q12.sh  
stud@debian:~/48A$ cat Kerala.txt  
Kerala is known as God's own country.  
A state in India.  
Kerala is famous for its unique geography, tranquil backwaters, unspoiled beaches, art forms and spices.  
It is also famous for its charming houseboats, sprawling tea plantations, unique eco-tourism, magnificent architecture, Ayurvedic treatments and unforgettable culinary experiences.  
stud@debian:~/48A$ bash q12.sh and Kerala.txt  
Kerala.txt  
-----  
Kerala is known as God's own country.  
A state in India.  
stud@debian:~/48A$ █
```

13. Write a shell script that computes the gross salary of an employee according to the following:

1) if basic salary is <1500 then HRA 10% of the basic and DA =90% of the basic

2) if basic salary is >1500 then HRA 500 and DA =98% of the basic

The basic salary is entered interactively through the key board

### **Shell Script**

```
echo "Enter basic salary"
```

```
read sal
```

```
if [ $sal -lt 1500 ]
```

```
then
```

```
    hra=$((sal * 10 / 100))
```

```
    da=$((sal * 90 / 100))
```

```
    gp=$((hra + da))
```

```
else
```

```
    hra=500
```

```
    da=$((sal * 98 / 100))
```

```
    gp=$((hra + da))
```

```
fi
```

```
echo "HRA : $hra"
```

```
echo "DA : $da"
```

```
echo "Gross Salary : $gp"
```

### Output

```
stud@debian:~$ vi salary.sh
stud@debian:~$ bash salary.sh
Enter basic salary
25000
HRA : 500
DA : 24500
Gross Salary : 25000
stud@debian:~$ █
```

## **EXPERIMENT-5**

### **INSTALLING LAMP ON UBUNTU**

#### **Step 1: Update Package Repository Cache**

Before you begin:

1. Open the terminal either by using the **CTRL+ALT+T** keyboard shortcut or by searching for the word *terminal* in **Ubuntu**
2. Make sure to update the package repository cache to ensure it installs the latest versions of the software. To do so, type in the following command:

**sudo apt-get update**

```
hp@hp-HP-Laptop-15s-du0xxx:~$ sudo apt-get update
Get:1 http://in.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Hit:2 https://screenrec.com/download/ubuntu stable InRelease
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [27
.6 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata
[61.0 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata
[2,464 B]
Get:1 http://in.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:1 http://in.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:1 http://in.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Hit:7 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:8 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 463 kB in 2min 36s (2,977 B/s)
Reading package lists... Done
hp@hp-HP-Laptop-15s-du0xxx:~$
```

## Step 2: Install Apache

1. To install Apache, run the following command in the terminal:

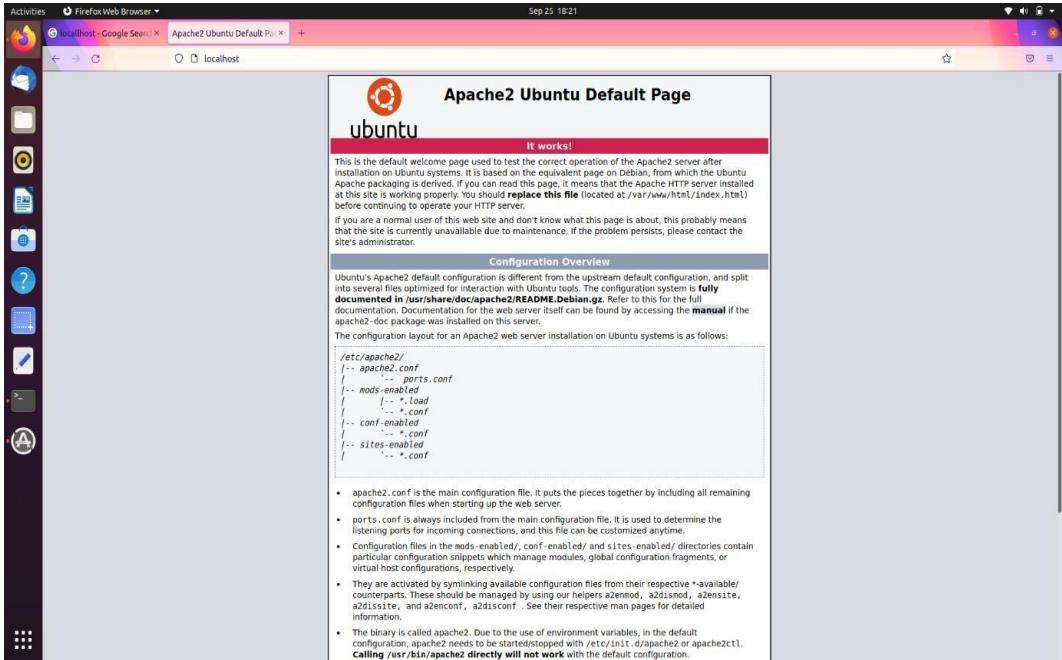
```
sudo apt-get install apache2
```

```
Reading package lists... Done
hp@hp-HP-Laptop-15s-du0xxx:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  enchant geolip-database libbind9-161 libboost-filesystem1.67.0
  libboost-iostreams1.67.0 libdns-export1107 libdns1107 libdns1109 libenchant1c2a
  libev12-14 libfprint0 libgeoip1 libgspell-1-1 libgutenprint-common
  libgutenprint9 libiptc0 libirs161 libisc-export1104 libisc1104 libisc1105
  libisccc161 libisccfg163 liblvm9 liblwres161 libnfs12 liboauth0
  printer-driver-gutenprint python3-asn1crypto shim ubuntu-software
  ubuntu-system-service
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
0 upgraded, 9 newly installed, 0 to remove and 66 not upgraded.
Need to get 1,819 kB of archives.
After this operation, 7,938 kB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

Press **y** (yes) and hit **ENTER** to permit the installation.

2. To ensure Apache is running, enter the Localhost of your server in the address bar and press **ENTER**.

The test Apache web server page should display as below.



### Step 3: Install PHP

1. To install PHP, run the following command:

```
$ sudo apt-get install php7.4
```

```
np@hp-HP-Laptop-15s-dwxxx:~$ sudo apt-get install php7.4
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  enchant geoip-database libbbin9-161 libboost filesystem1.67.0 libboost iostreams1.67.0 libdns-export1:107 libdns1:109 libenchant2ca libxiv2-14 libfprint0 libgeoip1 libgspell-1-1
  libgutenprint-common libgutenprint9 libiptc0 libirs161 libisc-export1:104 libisc1:105 libisccfg163 libl1w9 liblres161 libnfs12 liboauth0 printer-driver-gutenprint
  python3-asn1crypto shin ubuntu-software ubuntu-system-service
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libapache2-mod-php7.4 php-common php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php7.4 php-common php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline
0 upgraded, 8 newly installed, 0 to remove and 66 not upgraded.
Need to get 4,015 kB of archives.
After this operation, 18.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

Press **y** and **ENTER** to allow the installation.

#### Step 4: Restart Apache

After the php installation you must restart the Apache service.

Enter the command:

```
$ sudo /etc/init.d/apache2 restart
```

#### Step 5: Test PHP Processing on Web Server

1. Create a basic **PHP script** and save it to the “web root” directory. This is necessary for Apache to find and serve the file correctly. This directory is located at **/var/www/html/**.

To create a file in that directory, type in the following command:

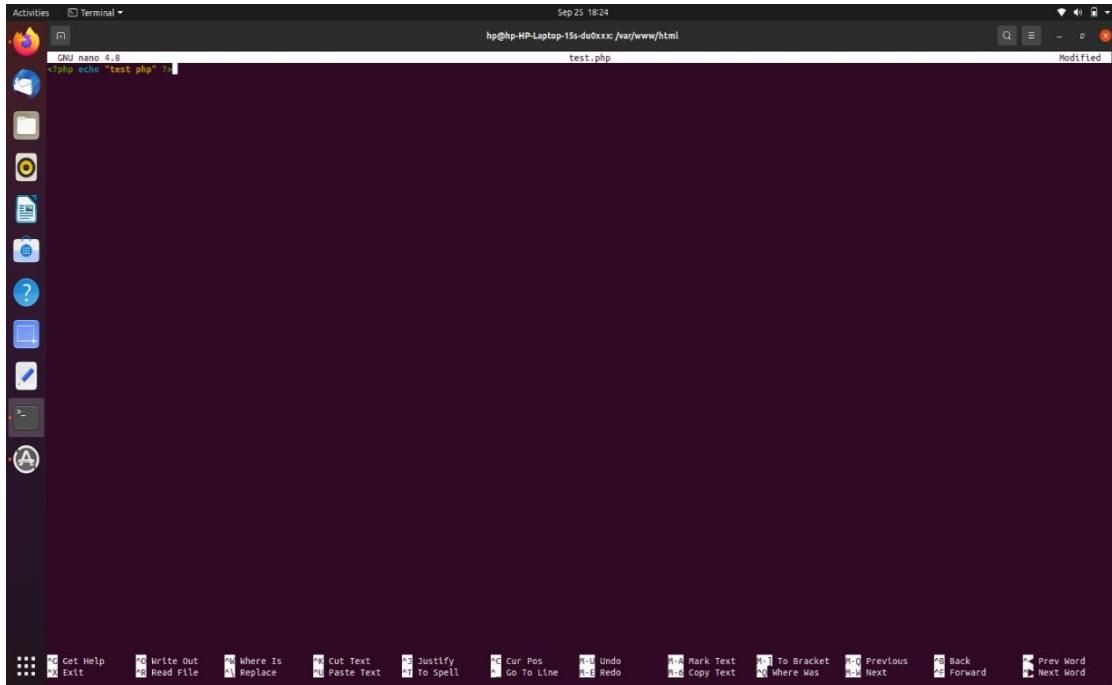
```
sudo nano /var/www/html/test.php
```

```
hp@hp-HP-Laptop-15s-du0xxx:~$ sudo nano /var/www/html/test.php
[sudo] password for hp:
hp@hp-HP-Laptop-15s-du0xxx:~$
```

This command opens the **bank file**.

2. Inside the file, type in the valid PHP code:

```
<?php
    Echo " test php ";
?>
```



A screenshot of a Linux desktop environment. On the left is a dock with icons for various applications like a browser, file manager, and system tools. In the center is a terminal window titled 'Terminal'. The terminal shows a command-line interface with the following text:  
Sep 25 18:24  
hp@hp-HP-Laptop-15s-du0xxx: /var/www/html  
test.php  
GNU nano 4.8  
<?php echo "test php";?>  
The status bar at the bottom of the terminal window indicates 'Modified'.

3. Press **CTRL + X** to save and close the file. Press **y** and **ENTER** to confirm.
4. Then check the code are run currently in php. Open the browser and enter the IP address (localhost/test.php).

### Step 6: Install Mysql server

1. To install Mysql server, run the following command:

```
$ sudo apt-get install mysql-server
```

```
Other options.
hp@hp-HP-Laptop-15s-du0xxx:~$ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  enchant geoip-database libbind9-161 libboost-filesystem1.67.0
  libboost-iostreams1.67.0 libdns-export1107 libdns1107 libdns1109
  libenchant1c2a libexiv2-14 libfprint0 libgeoip1 libgspell-1-1
  libgutenprint-common libgutenprint9 libiptc0 libirs161 libisc-export1104
  libisc1104 libisc1105 libisccc161 libisccfg163 liblvm9 liblwres161 libnfs12
  liboauth0 printer-driver-gutenprint python3-asn1crypto shim ubuntu-software
  ubuntu-system-service
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
```

2. Then it's asking us for a root password. Enter the password. Again we get to repeat it

### Step 7: Check the Mysql server

1. To check Mysql server, run the following command

```
$ mysql -u root -p
```

- Enter the root password and press enter

```
hp@hp-HP-Laptop-15s-du0xxx:~$ mysql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
hp@hp-HP-Laptop-15s-du0xxx:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.0.26-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database testdb;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| testdb             |
+--------------------+
5 rows in set (0.00 sec)

mysql> 
```

## 2. Create a database testdb and show it

- Enter the command  
Create database  
testdb;  
  
Show databases;
- So mysql is working then exit the mysql prompt just enter **exit**;

## Step 8: Install PHP Myadmin

- To install PHP Myadmin, run the following command:

```
$ sudo apt-get install phpmyadmin
```

```
hp@hp-HP-Laptop-15s-du0xxx:~$ sudo apt-get install phpmyadmin
[sudo] password for hp:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  acf apg colord-data enchant geopip-database gnome-control-center-faces gnome-online-accounts gsfonts hplip-data libbind9-161 libboost-filesystem1.67.0 libboost-iostreams1.67.0 libcolor-gtk1
  libcolorhug2 libdns-export1107 libdns1107 libenchant1c2a libexiv2-14 libfprint0 libgeoip1 libgsound0 libgspell-1-1 libgsdp-1.2-0 libgupnp-1.2-0 libgupnp-av-1.6-2 libgupnp-dlna-2.0-3
  libgutenprint-common libgutenprint9 libieee1284-3 libimagequant0 libiptc0 libirs161 libisc-export104 libisc1104 libisccc161 libiscfg163 libl1w9 liblres161 libnfs12 liboauth0
  librygel-core-2.6-2 librygel-db-2.6-2 librygel-renderer-2.6-2 librygel-server-2.6-2 lib sane-common libsnmp-base libwebpnu3 mobile-broadband-provider-info network-manager-gnome
  printer-driver-gutenprint printer-driver-postscript-hp python3-asn1crypto python3-nacaroonbakery python3-olefile python3-pil python3-protobuf python3-pynacaroons python3-renderpm python3-reportlab
  python3-reportlab-accel python3-rfc3339 python3-tz rygel shim ubuntu-software ubuntu-system-service
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  dbconfig-common dbconfig-mysql icc-profiles-free javascript-common libjs-jquery libjs-openlayers libjs-sphinxdoc libjs-underscore libonig5 libzip5 php-bz2 php-curl php-gd php-google-recaptcha
  php-mbstring php-mysql php-phmyadmin-notranslator php-phmyadmin-shapefile php-phmyadmin-sql-parser php-psr-cache php-psr-container php-psr-log php-symfony-cache
  php-symfony-cache-contracts php-symfony-expression-language php-symfony-service-contracts php-symfony-var-exporter php-tcpdf php-twlg php-twlg-extensions php-xml php-zip php7.4-bz2 php7.4-curl
  php7.4-gd php7.4-mbstring php7.4-mysql php7.4-xnl php7.4-zip
Suggested packages:
  php-dbase php-libodium php-ncrypt php-gmp php-symfony-service-implementation php-imagick php-twlg-doc php-symfony-translation php-recode php-gd2 php-pragmarx-google2fa php-bacon-qr-code
  php-sylius-uzf-php-server
Recommended packages:
  php-ncrypt
The following NEW packages will be installed:
  dbconfig-common dbconfig-mysql icc-profiles-free javascript-common libjs-jquery libjs-openlayers libjs-sphinxdoc libjs-underscore libonig5 libzip5 php-bz2 php-curl php-gd php-google-recaptcha
  php-mbstring php-mysql php-phmyadmin-notranslator php-phmyadmin-shapefile php-phmyadmin-sql-parser php-psr-cache php-psr-container php-psr-log php-symfony-cache
  php-symfony-cache-contracts php-symfony-expression-language php-symfony-service-contracts php-symfony-var-exporter php-tcpdf php-twlg php-twlg-extensions php-xml php-zip php7.4-bz2 php7.4-curl
  php7.4-gd php7.4-mbstring php7.4-mysql php7.4-xnl php7.4-zip phmyadmin
8 upgraded, 41 newly installed, 0 to remove and 61 not upgraded.
Need to get 16.0 MB of archives.
After this operation, 71.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

Press **y** and **ENTER** to allow the installation

- Then its ask what type of server, we have Apache2 is set by default that's what we want then press ok
- Then a configuration prompt are open . here we're going to just choose yes and then it ask the input password for phpmyadmin
- Then check it correct . go to the localhost/phpmyadmin. Here we can not found it so

We have to actually edit the file php is located in Apache2 folder.

- Enter the following command to edit the file

```
$ sudo nano/etc/php7.4/apache2.php.ini
```

- Then we need to uncomment an **extension=mysql.so**. find it the file just remove the Semicolon.

6. Then enter **ctl+x** to save

### Step 9: Restart Apache

After the php installation you must restart the Apache service.

Enter the command:

```
$ sudo /etc/init.d/apache2 restart
```

### Step 9.1: Include phpmyadmin in apache configuration

1. Enter the command:

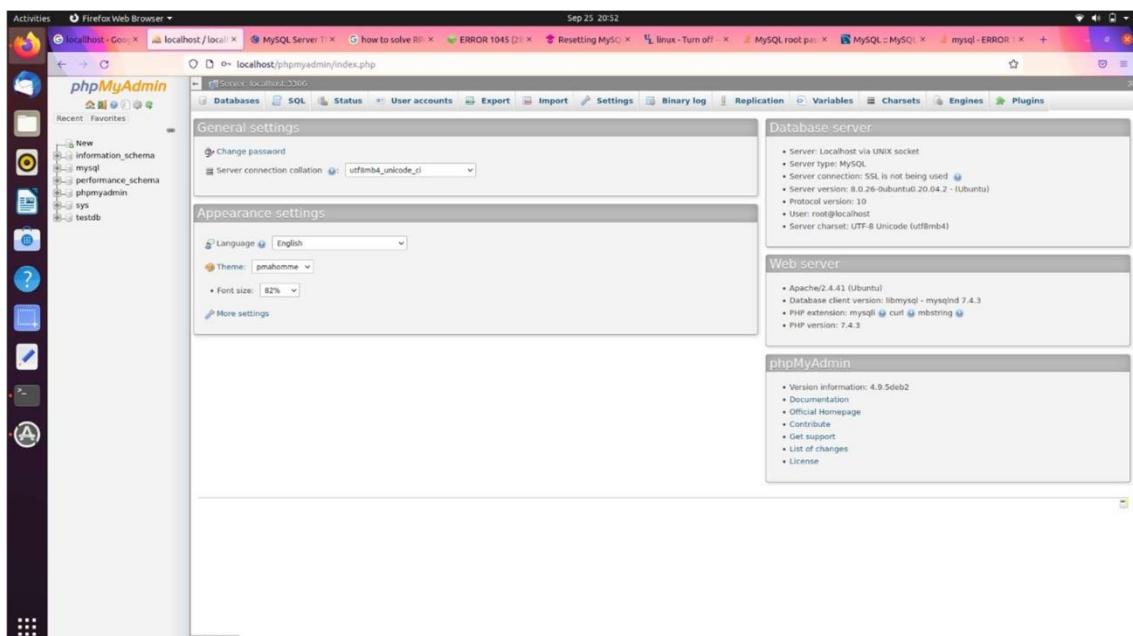
```
$ sudo nano/etc/apache2/apache2.conf
```

2. Type the following command to the nano editor

**Include /etc/phpmyadmin/apache.conf**

3. Then enter **ctl+x** to save

4. Then again restart the apache



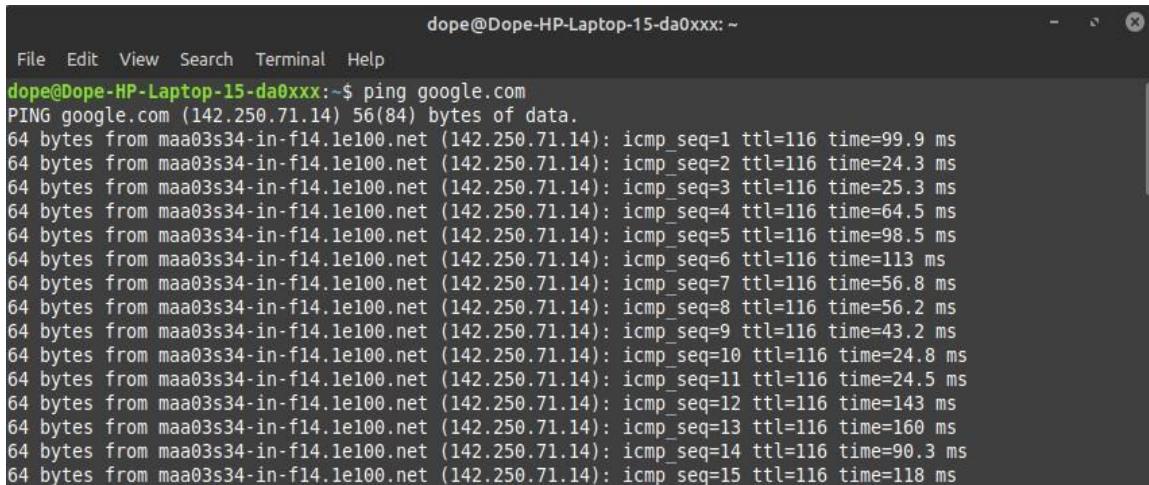
## EXPERIMENT-6

### NETWORKING COMMANDS

#### **PING COMMAND**

PING (Packet Internet Groper) command is used to check the network connectivity between host and server/host. This command takes as input the IP address or the URL and sends a data packet to the specified address with the message “PING” and get a response from the server/host this time is recorded which is called latency. Fast ping low latency means faster connection. Ping uses ICMP(Internet Control Message Protocol) to send an ICMP echo message to the specified host if that host is available then it sends ICMP reply message. Ping is generally measured in millisecond every modern operating system has this ping pre-installed.

#### **Syntax: ping [OPTIONS] DESTINATION**

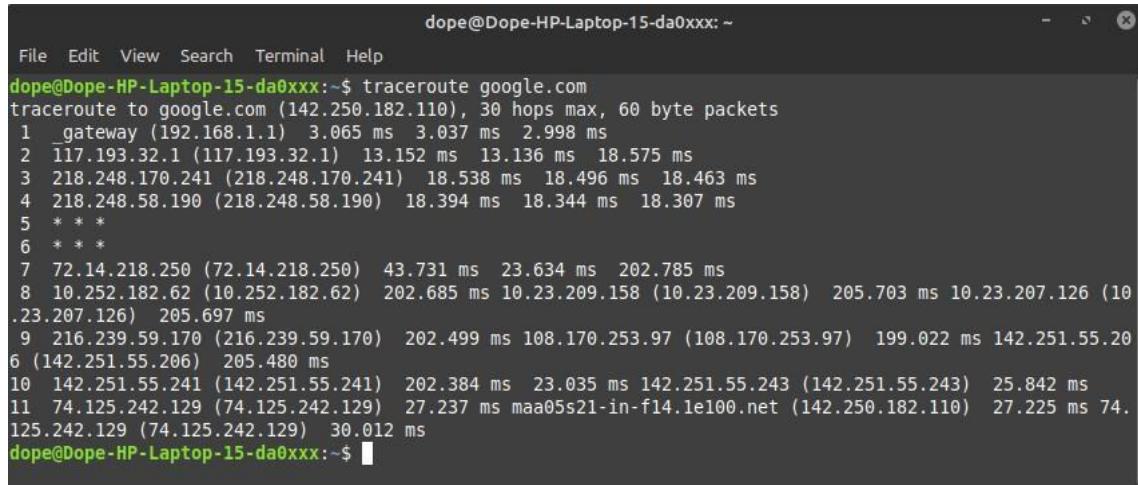


```
dope@Dope-HP-Laptop-15-da0xxx:~$ ping google.com
PING google.com (142.250.71.14) 56(84) bytes of data.
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=1 ttl=116 time=99.9 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=2 ttl=116 time=24.3 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=3 ttl=116 time=25.3 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=4 ttl=116 time=64.5 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=5 ttl=116 time=98.5 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=6 ttl=116 time=113 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=7 ttl=116 time=56.8 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=8 ttl=116 time=56.2 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=9 ttl=116 time=43.2 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=10 ttl=116 time=24.8 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=11 ttl=116 time=24.5 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=12 ttl=116 time=143 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=13 ttl=116 time=160 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=14 ttl=116 time=90.3 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=15 ttl=116 time=118 ms
```

## **TRACEROUTE COMMAND**

**Traceroute** command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes. Below image depicts how traceroute command is used to reach the Google (172.217.26.206) host from the local machine and it also prints detail about all the hops that it visits in between.

**Syntax: traceroute [options] host\_Address [pathlength]**



```
dope@Dope-HP-Laptop-15-da0xxx:~$ traceroute google.com
traceroute to google.com (142.250.182.110), 30 hops max, 60 byte packets
 1 _gateway (192.168.1.1)  3.065 ms  3.037 ms  2.998 ms
 2 117.193.32.1 (117.193.32.1)  13.152 ms  13.136 ms  18.575 ms
 3 218.248.170.241 (218.248.170.241)  18.538 ms  18.496 ms  18.463 ms
 4 218.248.58.190 (218.248.58.190)  18.394 ms  18.344 ms  18.307 ms
 5 * * *
 6 * * *
 7 72.14.218.250 (72.14.218.250)  43.731 ms  23.634 ms  202.785 ms
 8 10.252.182.62 (10.252.182.62)  202.685 ms  10.23.209.158 (10.23.209.158)  205.703 ms  10.23.207.126 (10
.23.207.126)  205.697 ms
 9 216.239.59.170 (216.239.59.170)  202.499 ms  108.170.253.97 (108.170.253.97)  199.022 ms  142.251.55.20
6 (142.251.55.206)  205.480 ms
10 142.251.55.241 (142.251.55.241)  202.384 ms  23.035 ms  142.251.55.243 (142.251.55.243)  25.842 ms
11 74.125.242.129 (74.125.242.129)  27.237 ms  maa05s21-in-f14.le100.net (142.250.182.110)  27.225 ms  74.
125.242.129 (74.125.242.129)  30.012 ms
dope@Dope-HP-Laptop-15-da0xxx:~$
```

## ROUTE COMMAND

**Route** command in Linux is used when you want to work with the IP/kernel routing table. It is mainly used to set up static routes to specific hosts or networks via an interface. It is used for showing or update the IP/kernel routing table

Syntax: route

```
dope@Dope-HP-Laptop-15-da0xxx:~$ route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
default         gateway        0.0.0.0       UG    600    0        0 wlo1
link-local      0.0.0.0       255.255.0.0   U     1000   0        0 wlo1
192.168.1.0    0.0.0.0       255.255.255.0 U     600    0        0 wlo1
dope@Dope-HP-Laptop-15-da0xxx:~$
```

## NSLOOKUP COMMAND

**Nslookup** stands for “Name Server Lookup”) is a useful command for getting information from DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS related problems.

Syntax: nslookup [option]

```
dope@Dope-HP-Laptop-15-da0xxx:~$ nslookup google.com
Server:  127.0.0.53
Address: 127.0.0.53#53

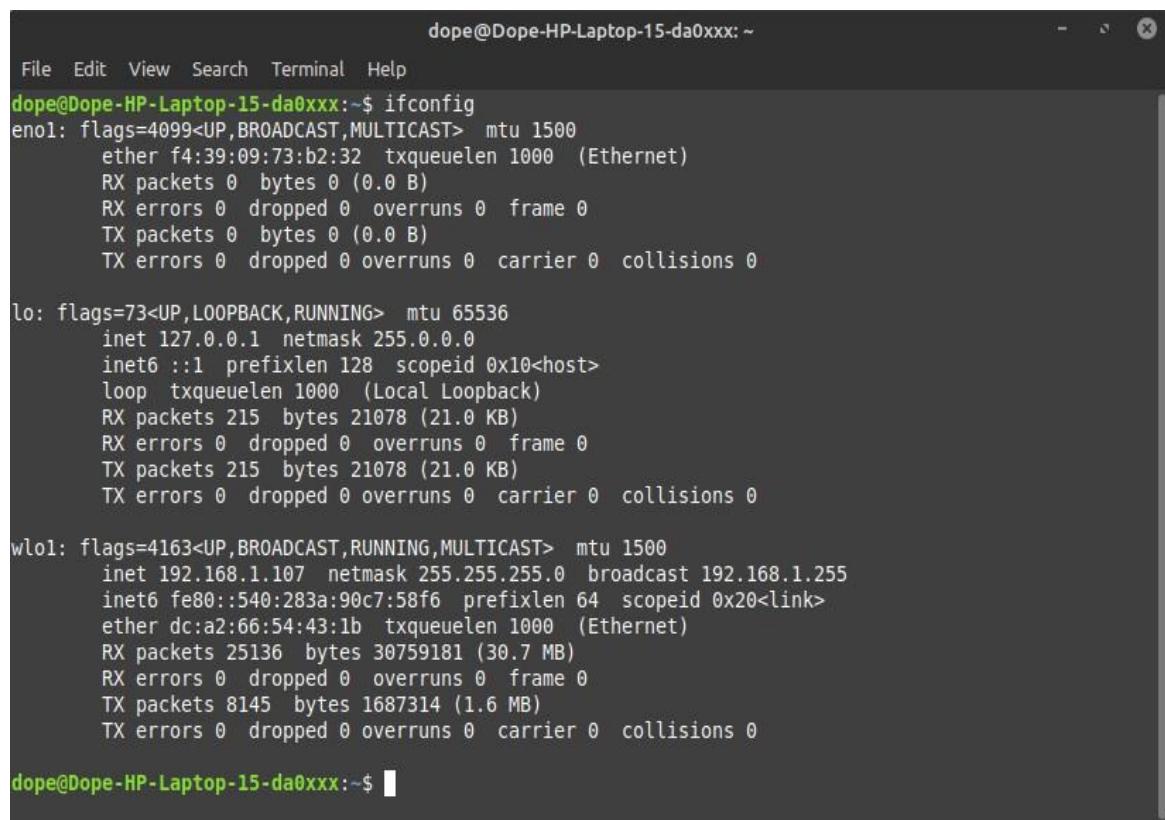
Non-authoritative answer:
Name: google.com
Address: 142.250.195.110
Name: google.com
Address: 2404:6800:4007:818::200e

dope@Dope-HP-Laptop-15-da0xxx:~$
```

## **IFCONFIG COMMAND**

**Ifconfig** (interface configuration) command is used to configure the kernel resident network interfaces. It is used at the boot time to set up the interfaces as necessary. After that, it is usually used when needed during debugging or when you need system tuning. Also, this command is used to assign the IP address and netmask to an interface or to enable or disable a given interface.

**Syntax:**    **ifconfig [..OPTIONS] [INTERFACE]**



```
dope@Dope-HP-Laptop-15-da0xxx:~
```

```
File Edit View Search Terminal Help
```

```
dope@Dope-HP-Laptop-15-da0xxx:~$ ifconfig
en0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether f4:39:09:73:b2:32 txqueuelen 1000  (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 215 bytes 21078 (21.0 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 215 bytes 21078 (21.0 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.107 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::540:283a:90c7:58f6 prefixlen 64 scopeid 0x20<link>
        ether dc:a2:66:54:43:1b txqueuelen 1000  (Ethernet)
        RX packets 25136 bytes 30759181 (30.7 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 8145 bytes 1687314 (1.6 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

dope@Dope-HP-Laptop-15-da0xxx:~$
```

## **EXPERIMENT-7**

### **INTRODUCTION TO VIRTUAL MACHINES**

#### **Creating a Virtual Machine**

##### **1. Install VirtualBox .**

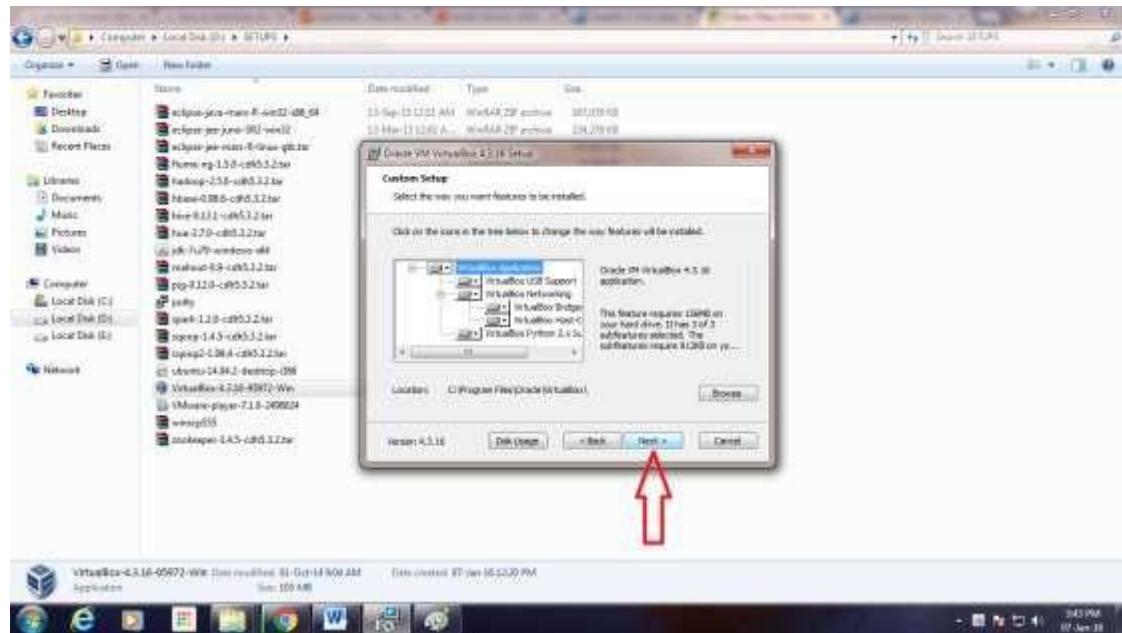
If you don't already have VirtualBox installed on your Windows or Mac computer, you'll need to install it before proceeding.

Following are the steps required to install VirtualBox(Oracle VM VirtualBox):

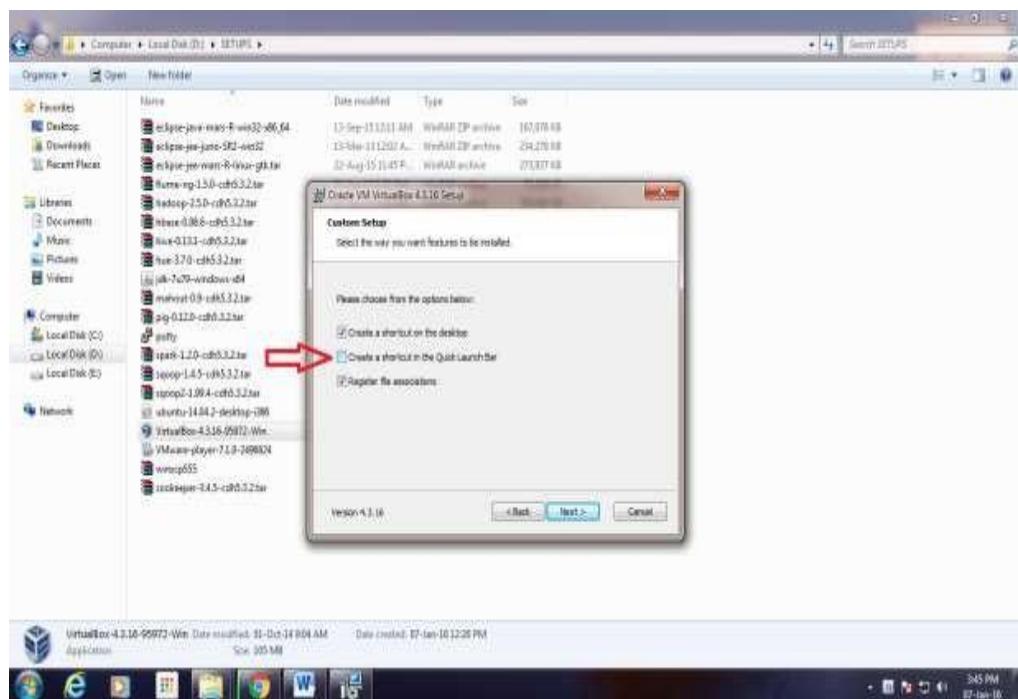
You can download the latest version of VirtualBox from the Virtual Box website:  
<https://www.virtualbox.org/wiki/Downloads> according to the version of your operating system Windows, Mac or Linux.

Click Next



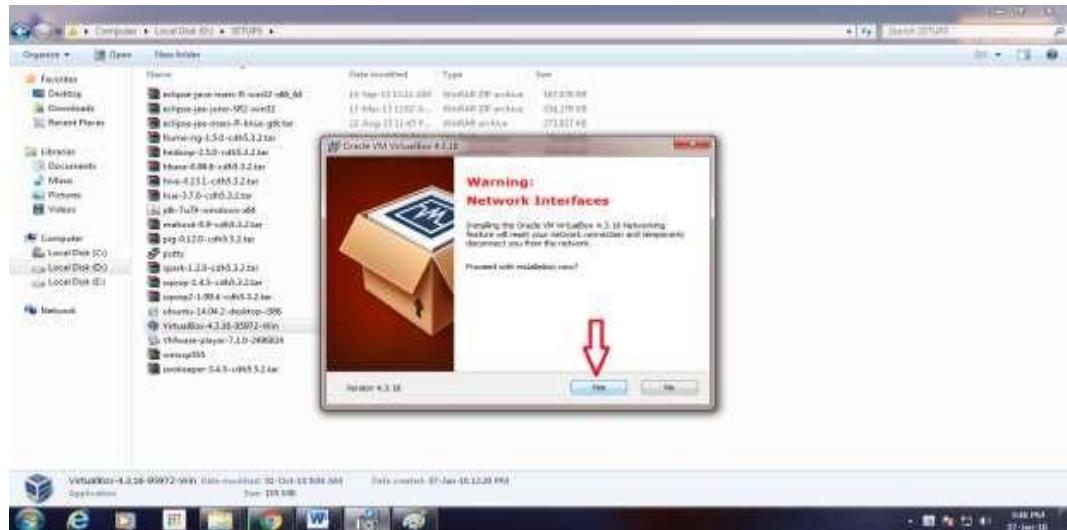
**To Install VirtualBox – Setup Wizard****Click Next****To Install VirtualBox – Custom Setup**

- Uncheck “Create a shortcut in the Quick Launch Bar” and click “Next”



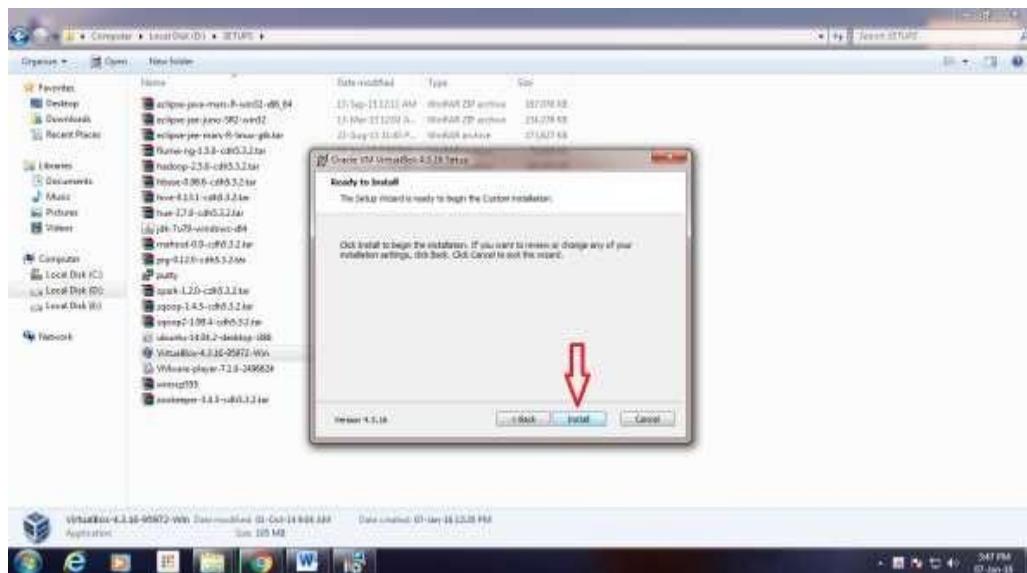
## To Install VirtualBox – Features Selection

- Click “Yes”



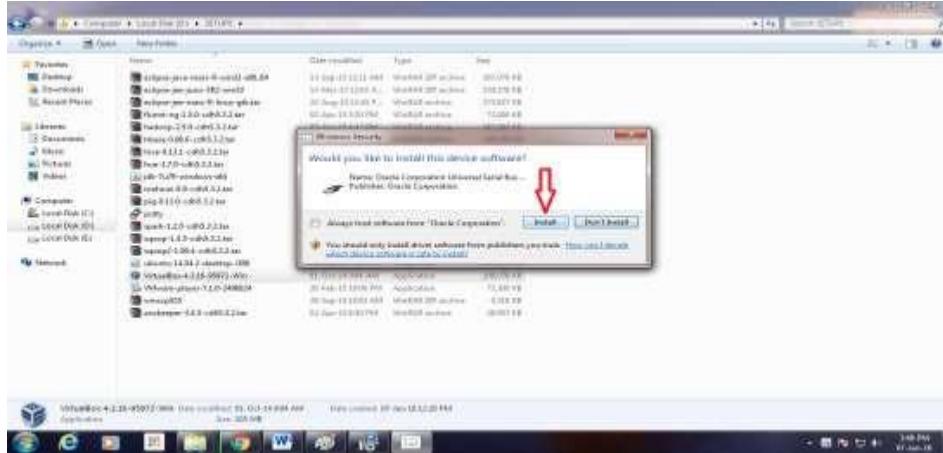
## To Install VirtualBox – Network Interfaces Warning

- Click “Install”



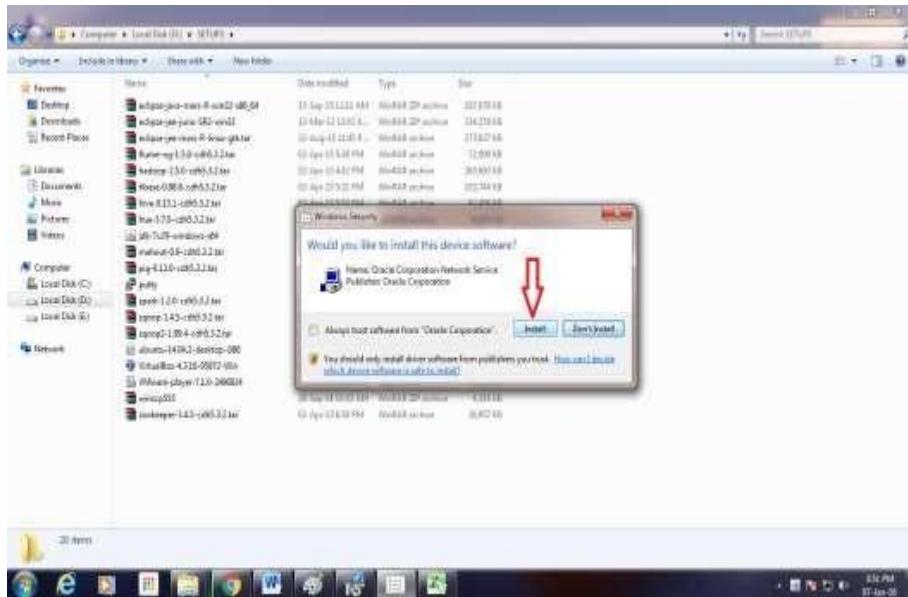
## Installation of Oracle VM VirtualBox – Ready to Install

- Click “Install”



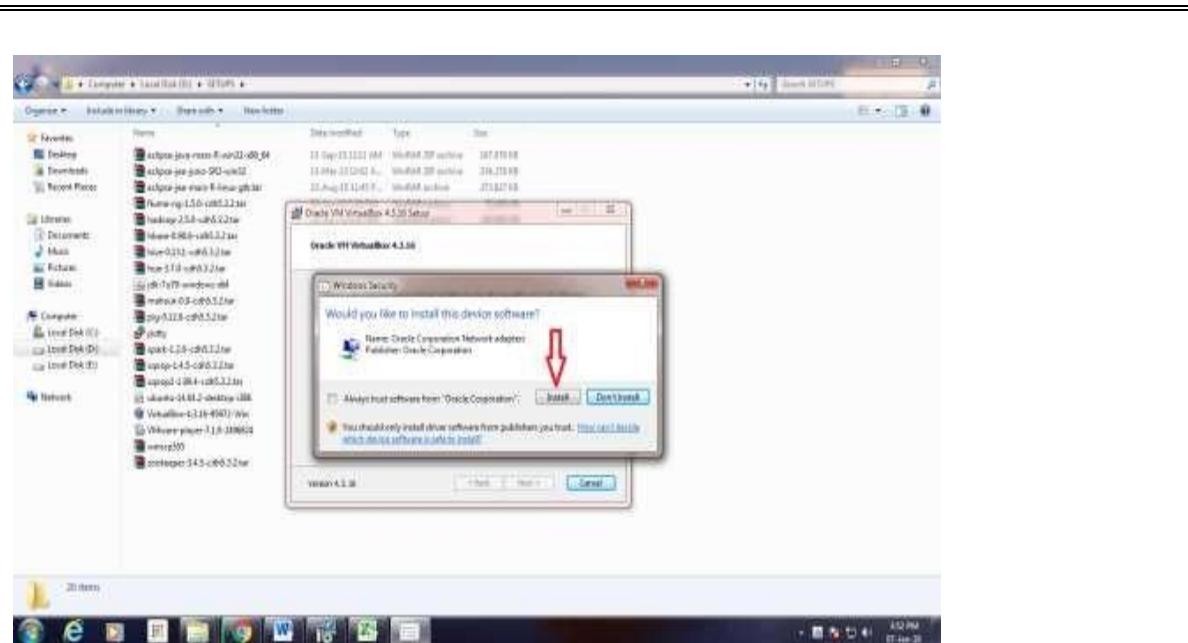
### Installation of Oracle VM VirtualBox- Serial Bus Software Installation

- Click “Install”



### Installation of Oracle VM VirtualBox – Network Service Installation

- Click “Install”



### Installation of Oracle VM VirtualBox – Network Adapters Installation

- Click “Finish”

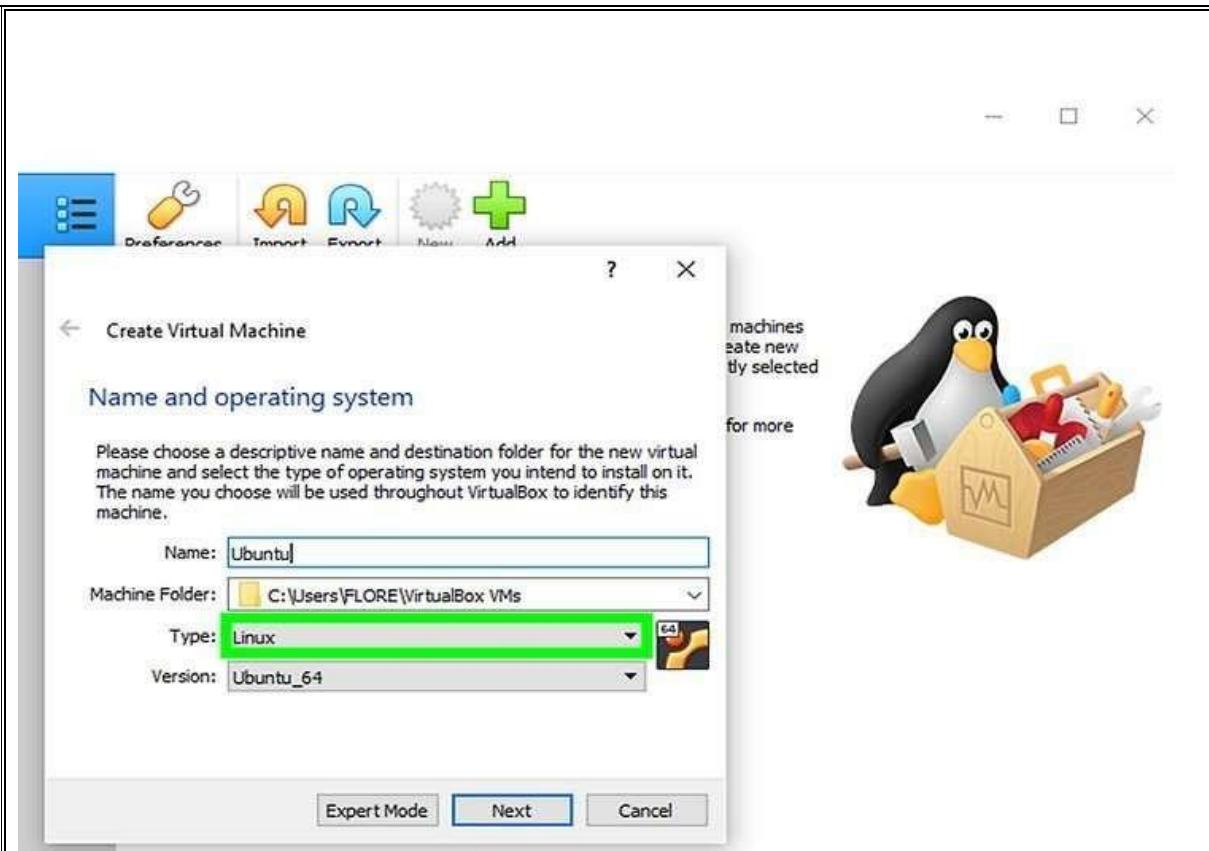


**Open VirtualBox.** Double-click (or click once on a Mac) the VirtualBox app icon.



**Click New.** It's a blue badge in the upper-left corner of the VirtualBox window. Doing so opens a pop-up menu.



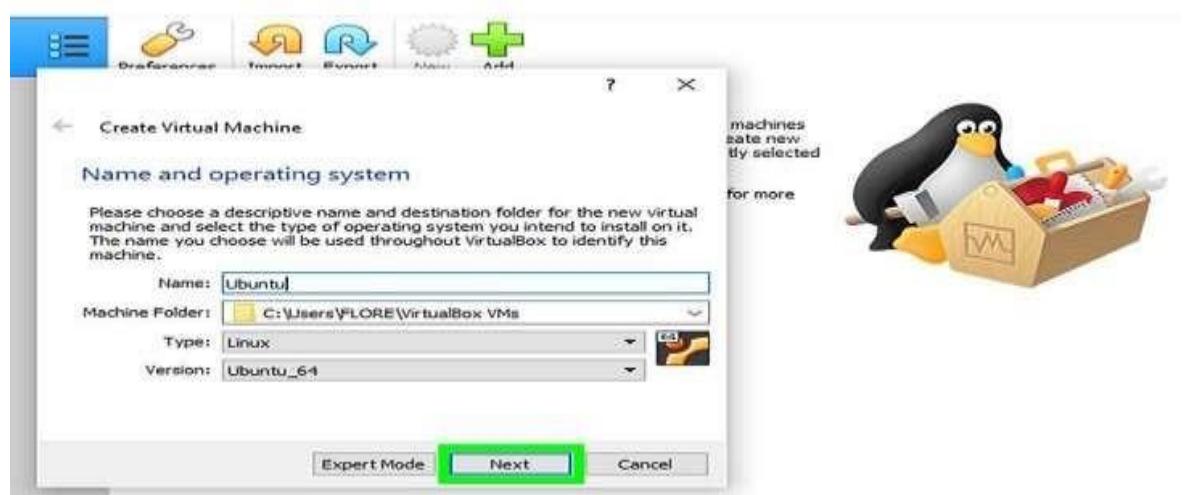


**Enter a name for your virtual machine.** Type whatever you want to name your virtual machine (e.g., Ubuntu) into the "Name" text field that's near the top of the pop-up menu.

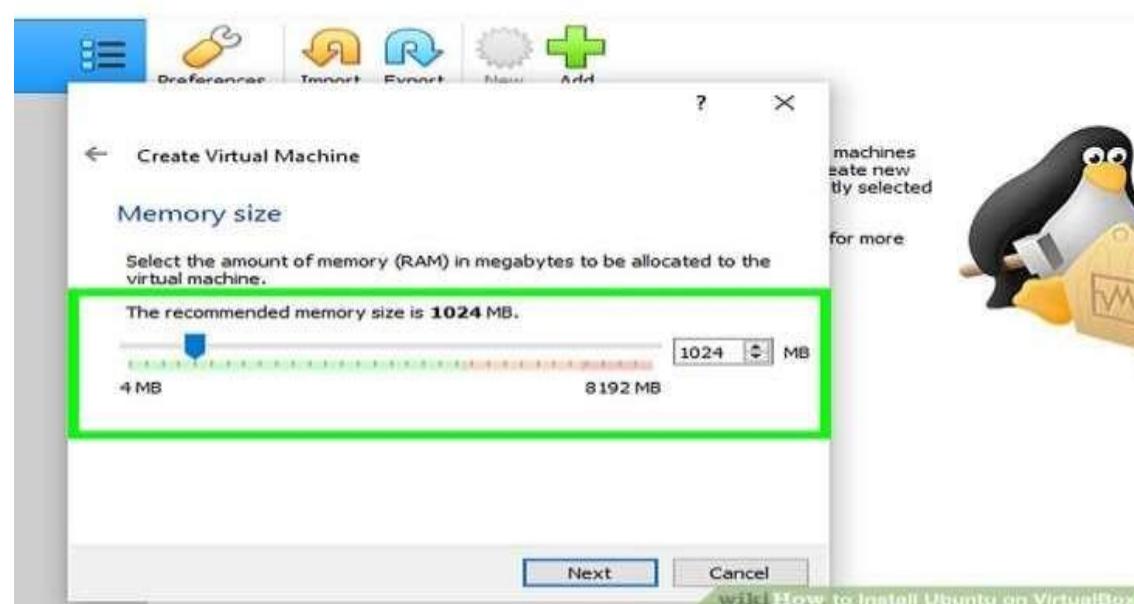
For **Operating System Type**, select the OS that you want to install.



Select the **version** of the operating system.



**Click Next.** It's at the bottom of the menu.



**Select an amount of RAM to use.** Click and drag the slider left or right to decrease or increase the amount of RAM that VirtualBox will have available for your virtual machine.

The ideal amount of RAM will automatically be selected when you get to this page.

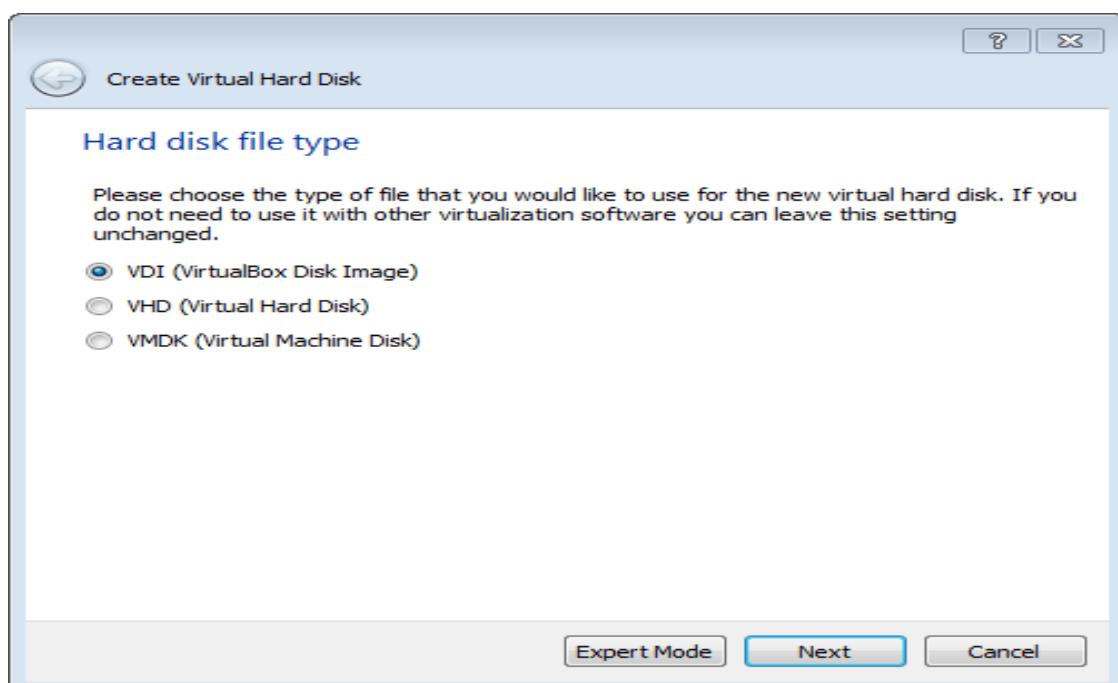
Make sure not to increase the RAM into the red section of the slider; try to keep the slider in the green.



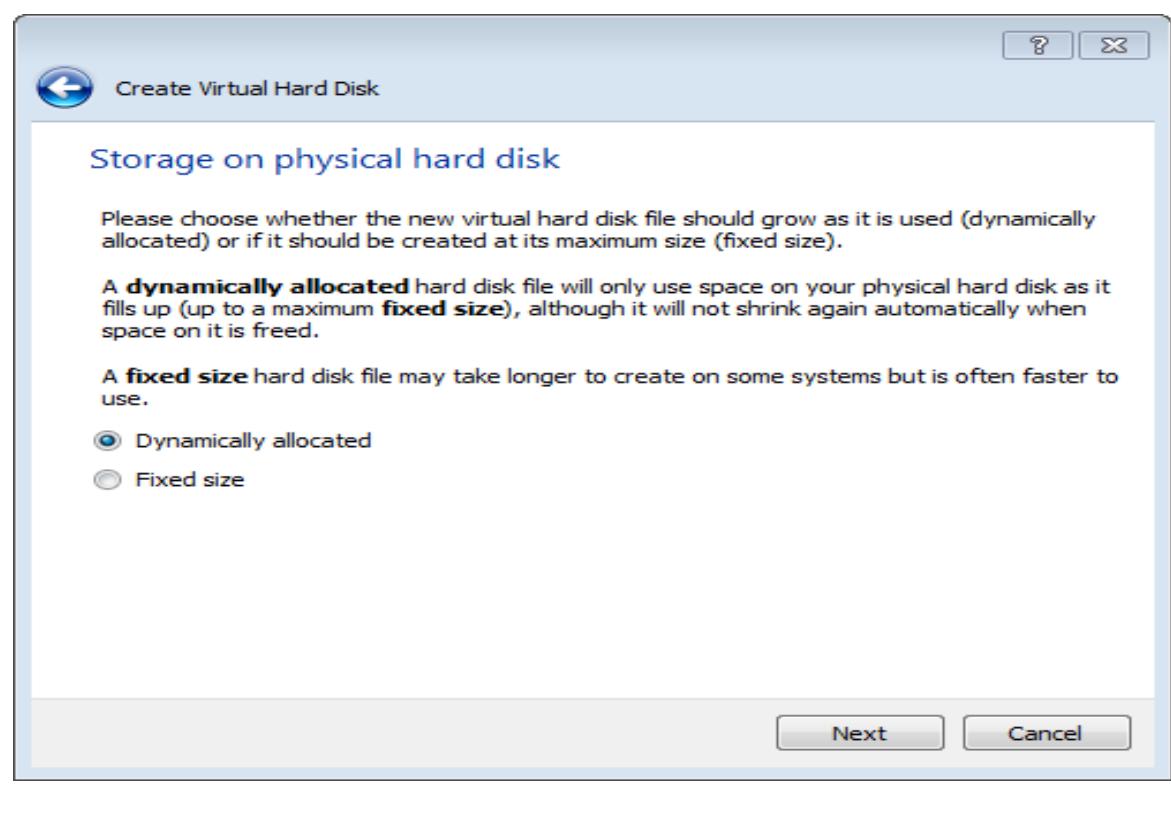
**Click Next.** It's at the bottom of the menu.



**Create your virtual machine's virtual hard drive.** The virtual hard drive is a section of your computer's hard drive space which will be used to store your virtual machine's files and programs:

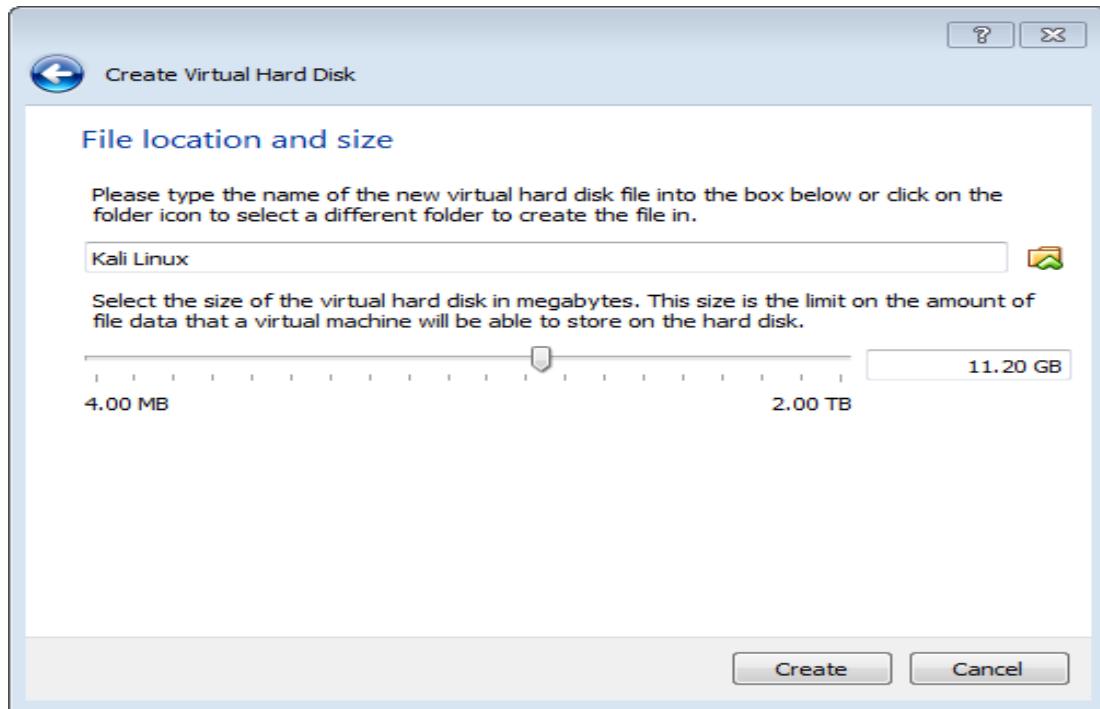


Use “VDI” to create a virtual hard disk



Choose “Dynamically allocated”

Allocate at Minimum 8 GB (recommended 10 or more).



Click **Create**, to create your new virtual machine. The virtual machine is displayed in the list on the left side of the VirtualBox Manager window, with the name that you entered initially.

VMs can run multiple operating system environments on a single physical computer, saving physical space, time and management costs.

## **EXPERIMENT-8**

### **WIRESHARK**

Wireshark is an open-source packet analyzer, which is used for education, analysis, software development, communication protocol development, and network troubleshooting. It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer, and network analyzer.

It is also used by network security engineers to examine security problems.

Wireshark is a data capturing program that "understands" the structure (encapsulation) of different networking protocols. It can parse and display the fields, along with their meanings as specified by different networking protocols. Wireshark uses pcap to capture packets, so it can only capture packets on the types of networks that pcap supports.

#### **Installation of Wireshark Software**

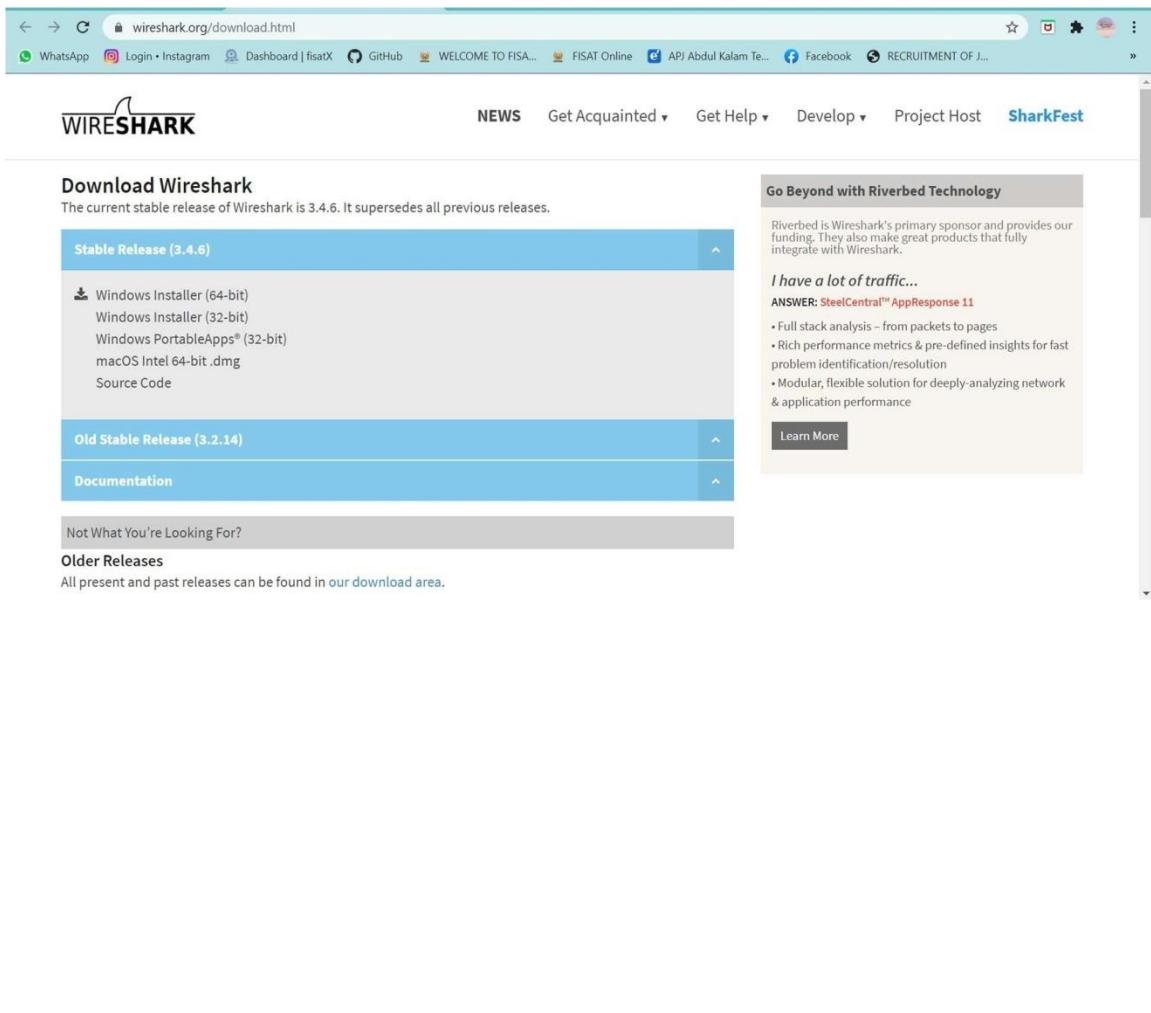
Downloading steps:-

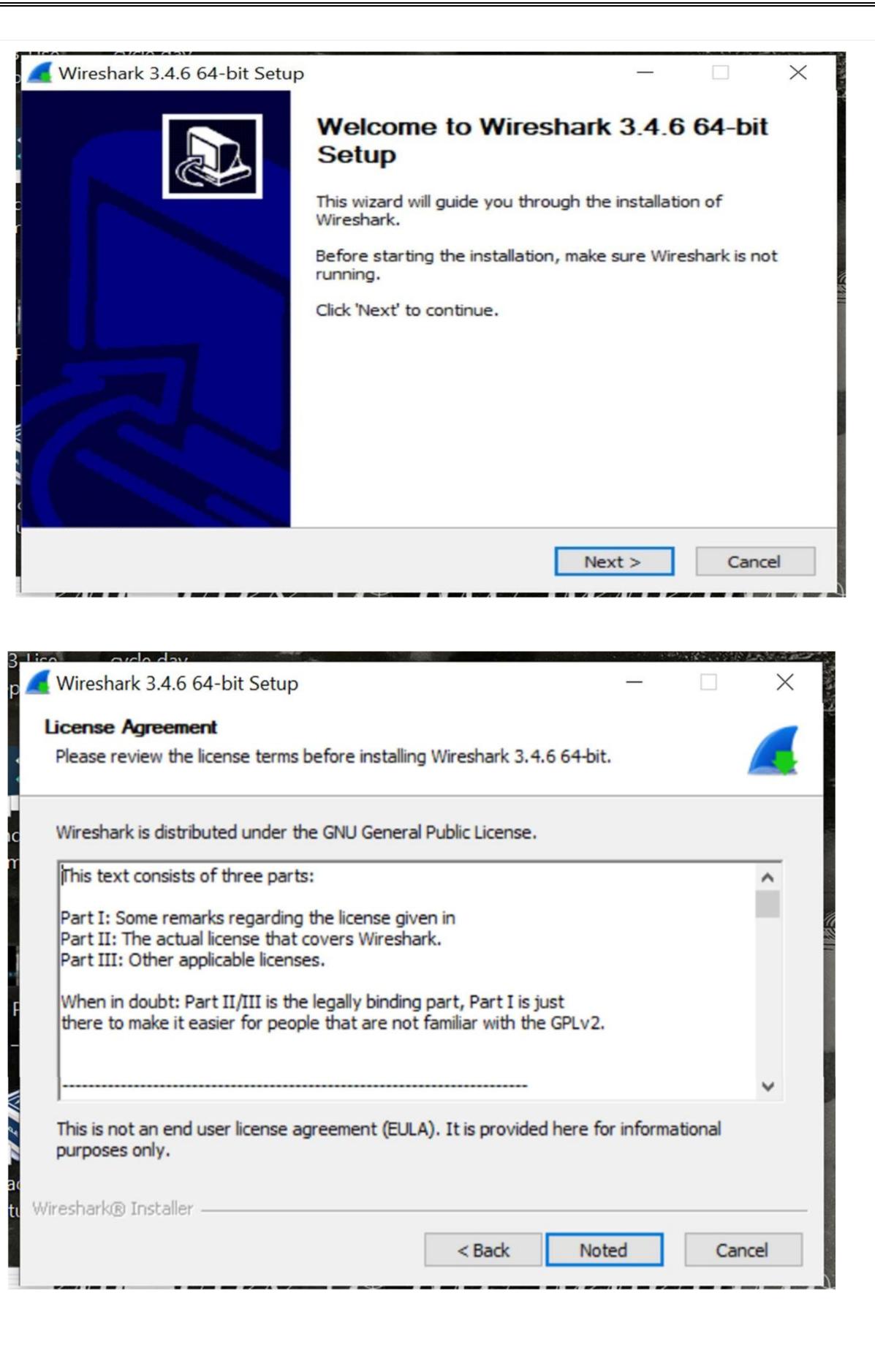
1. Open a web browser.
2. Navigate to <http://www.wireshark.org>.
3. Select Download Wireshark.
4. Select the Wireshark Windows Installer matching your system type. Save the program in the Downloads folder.
5. Close the web browser.

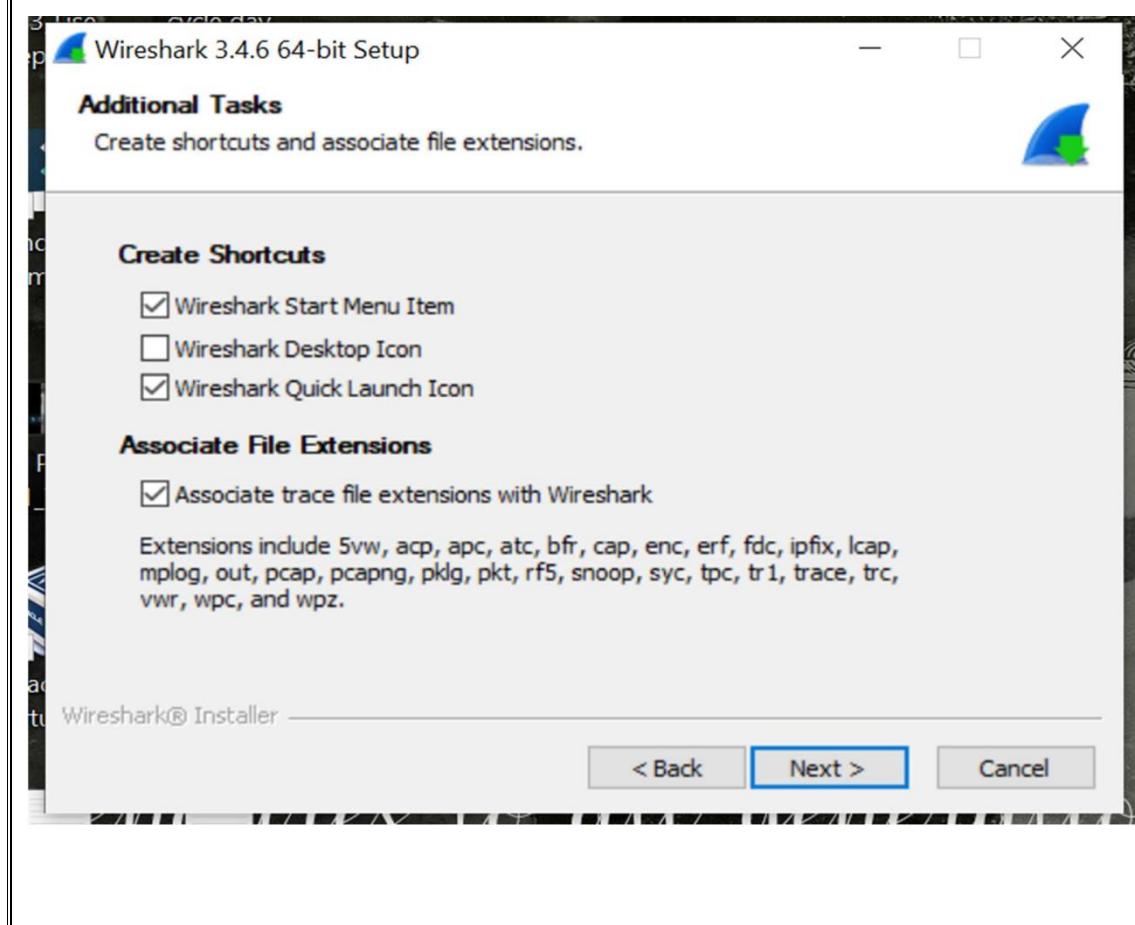
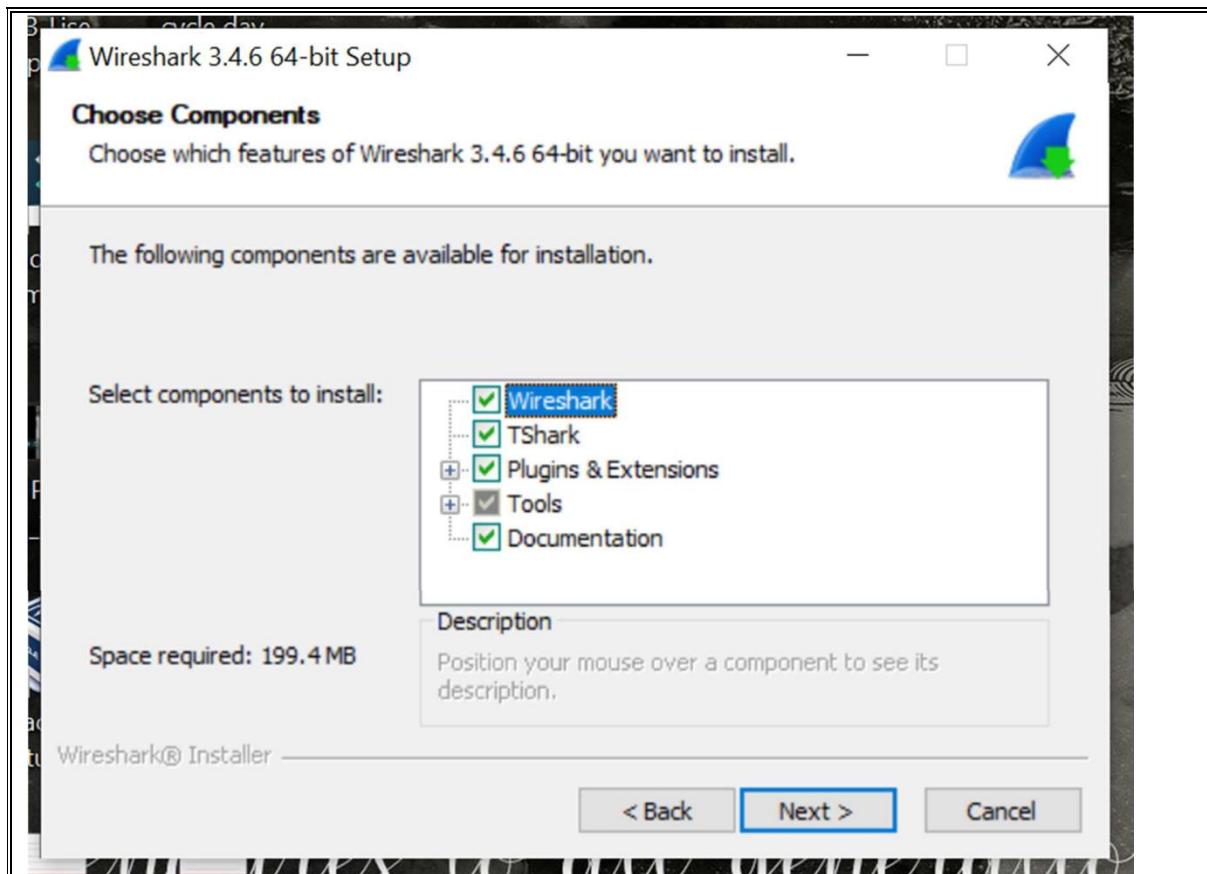
Installation process:-

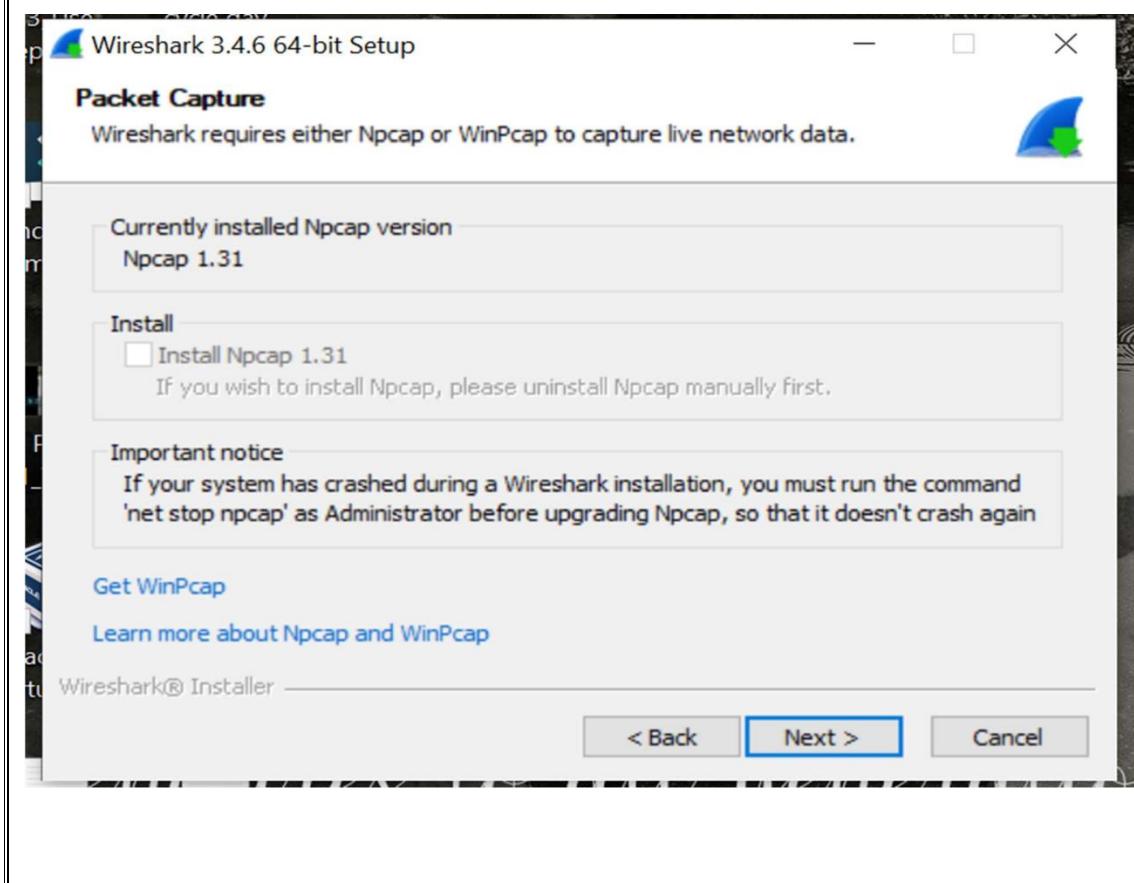
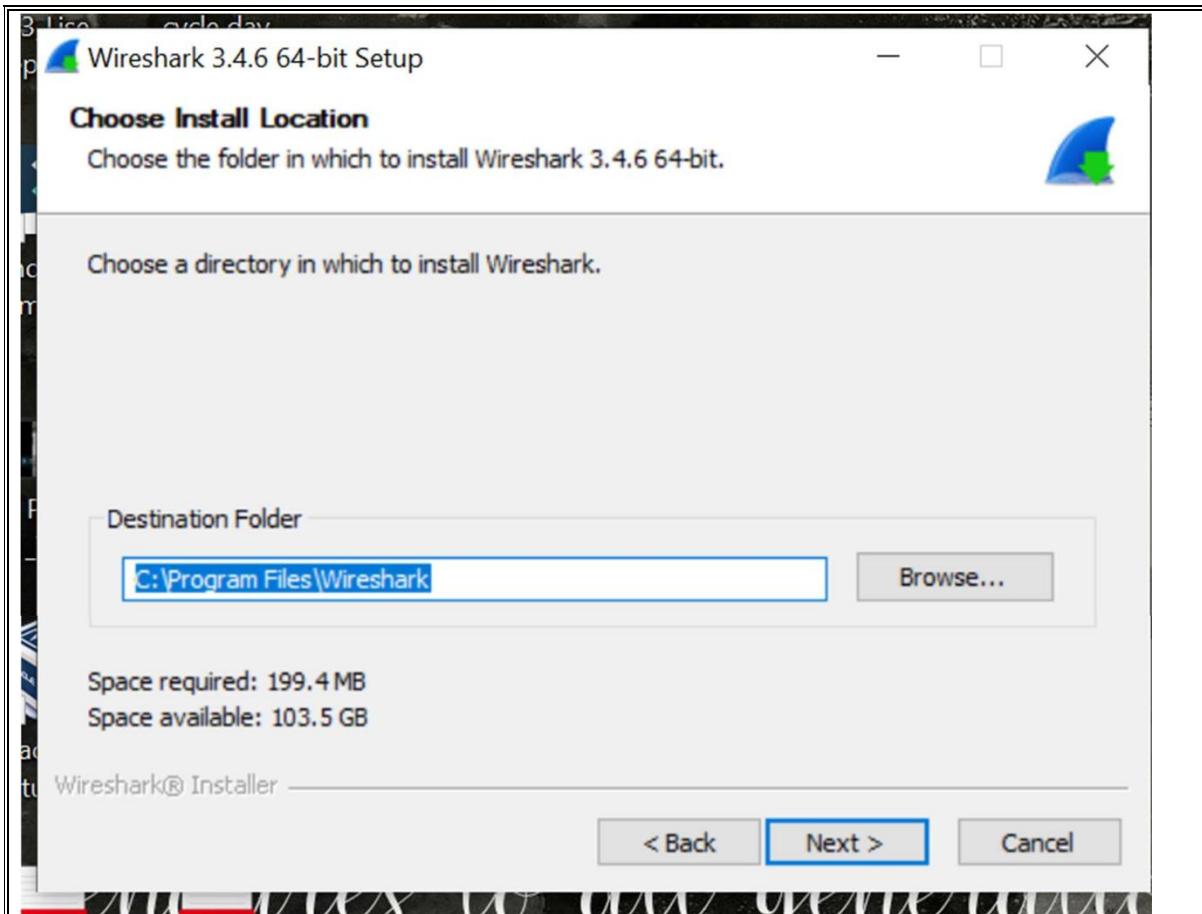
1. Double-click on the file to open it.
2. If you see a User Account Control dialog box, select Yes to allow the program to make changes to this computer.
3. Select Next to start the Setup Wizard.
4. Review the license agreement. If you agree, select I Agree to continue.

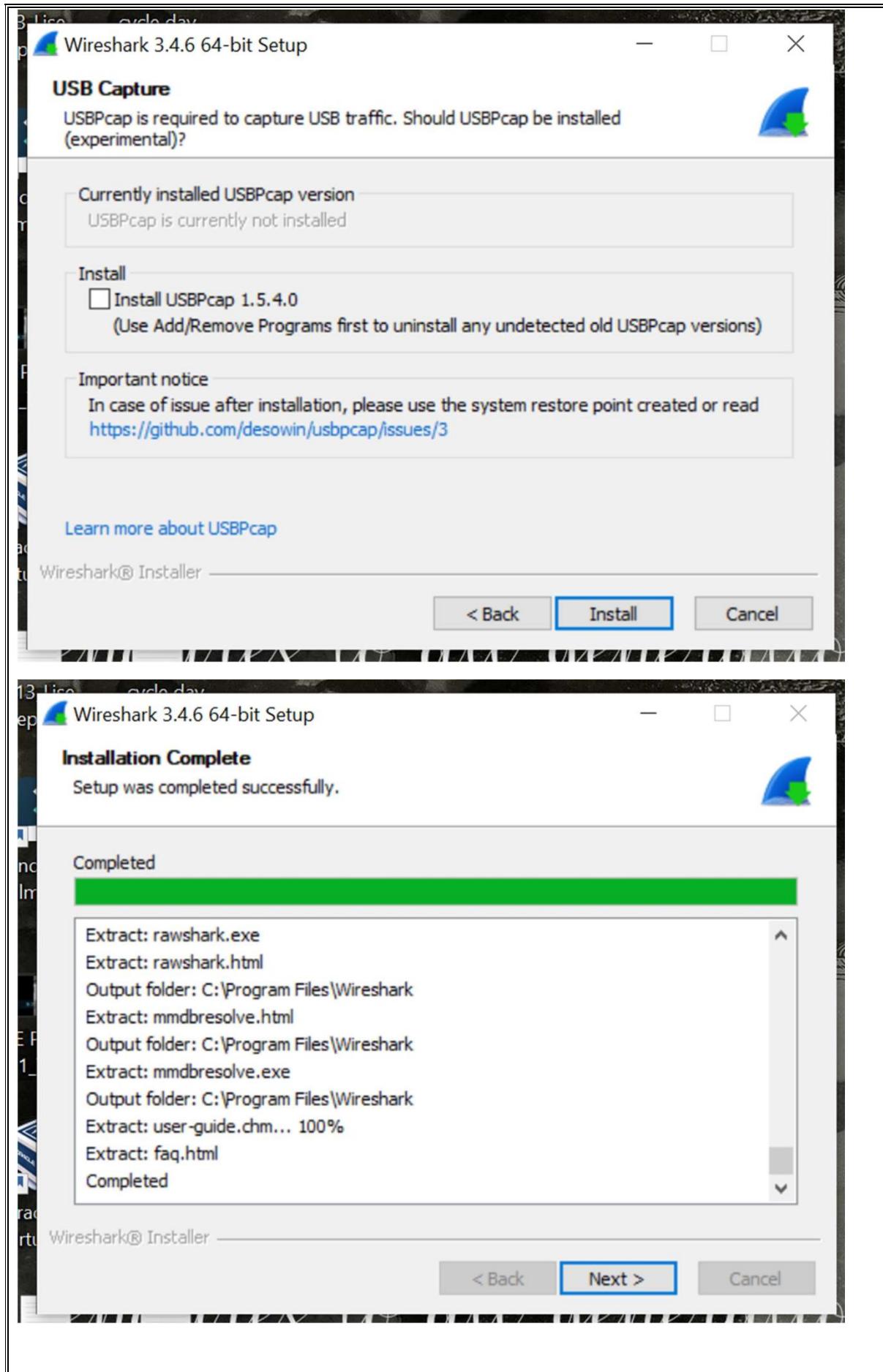
5. Select Next to accept the default components.
6. Select the shortcuts you would like to have created. Leave the file extensions selected. Select Next to continue.
7. Select Next to accept the default install location.
8. Select Install to begin installation.
9. Select Next to install WinPcap.
10. Select Next to start the Setup Wizard.
11. Review the license agreement. If you agree, select I Agree to continue.
12. Select Install to begin installation.
13. Select Finish to complete the installation of WinPcap.
14. Select Next to continue with the installation of Wireshark.
15. Select Finish to complete the installation of Wireshark.

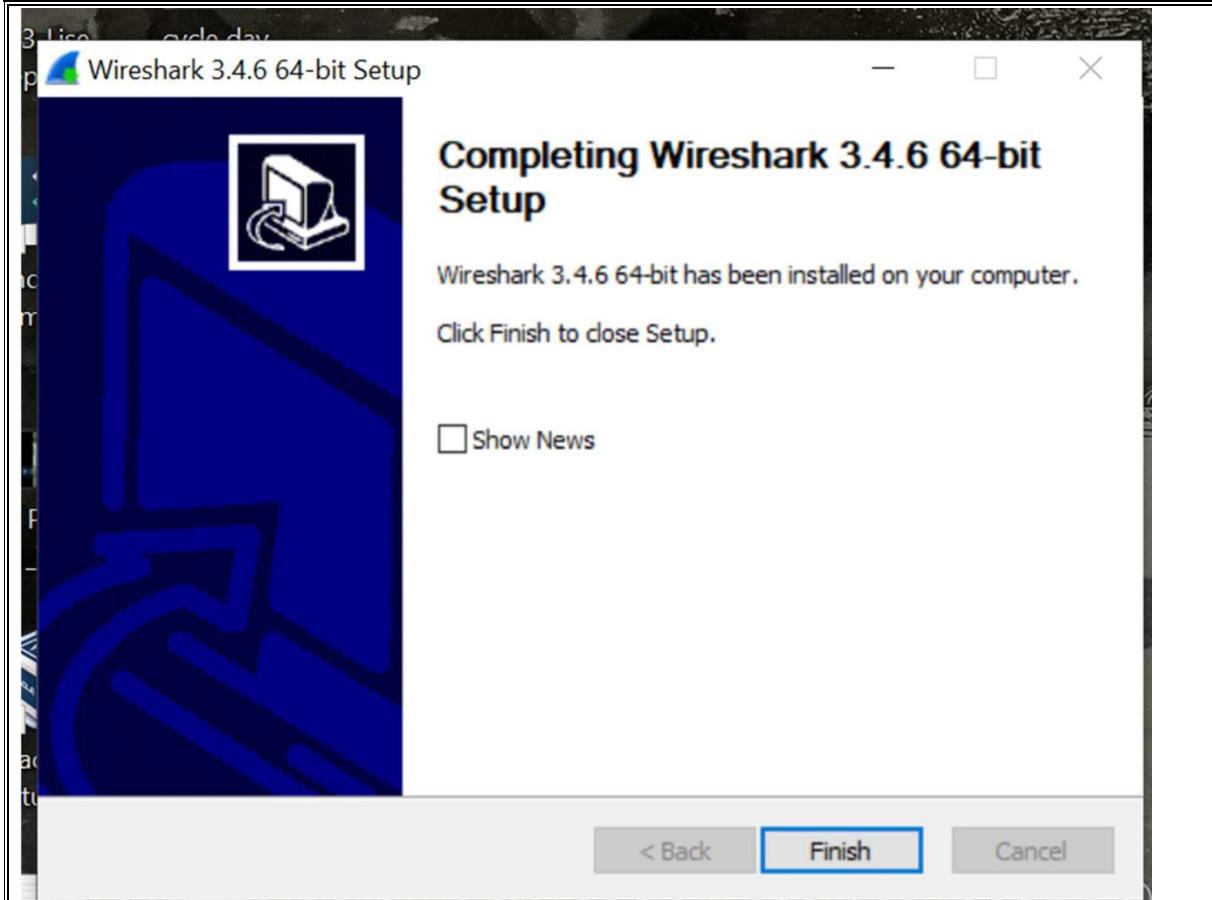




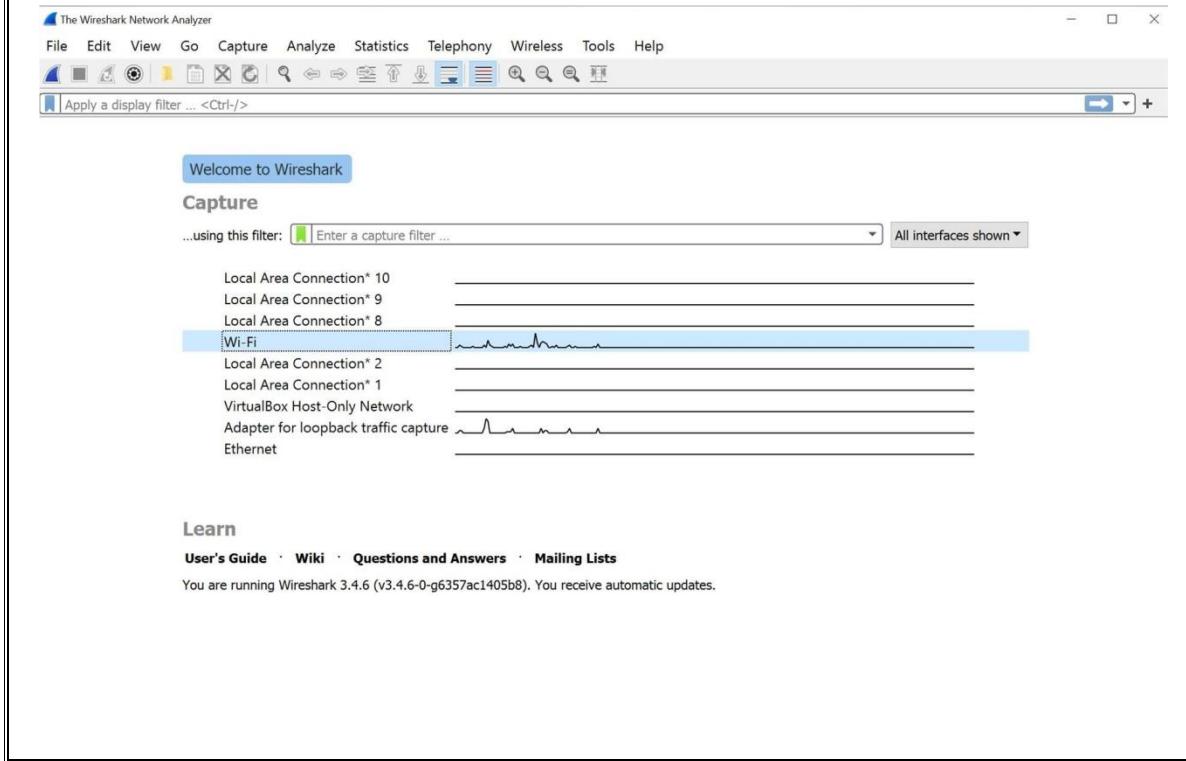




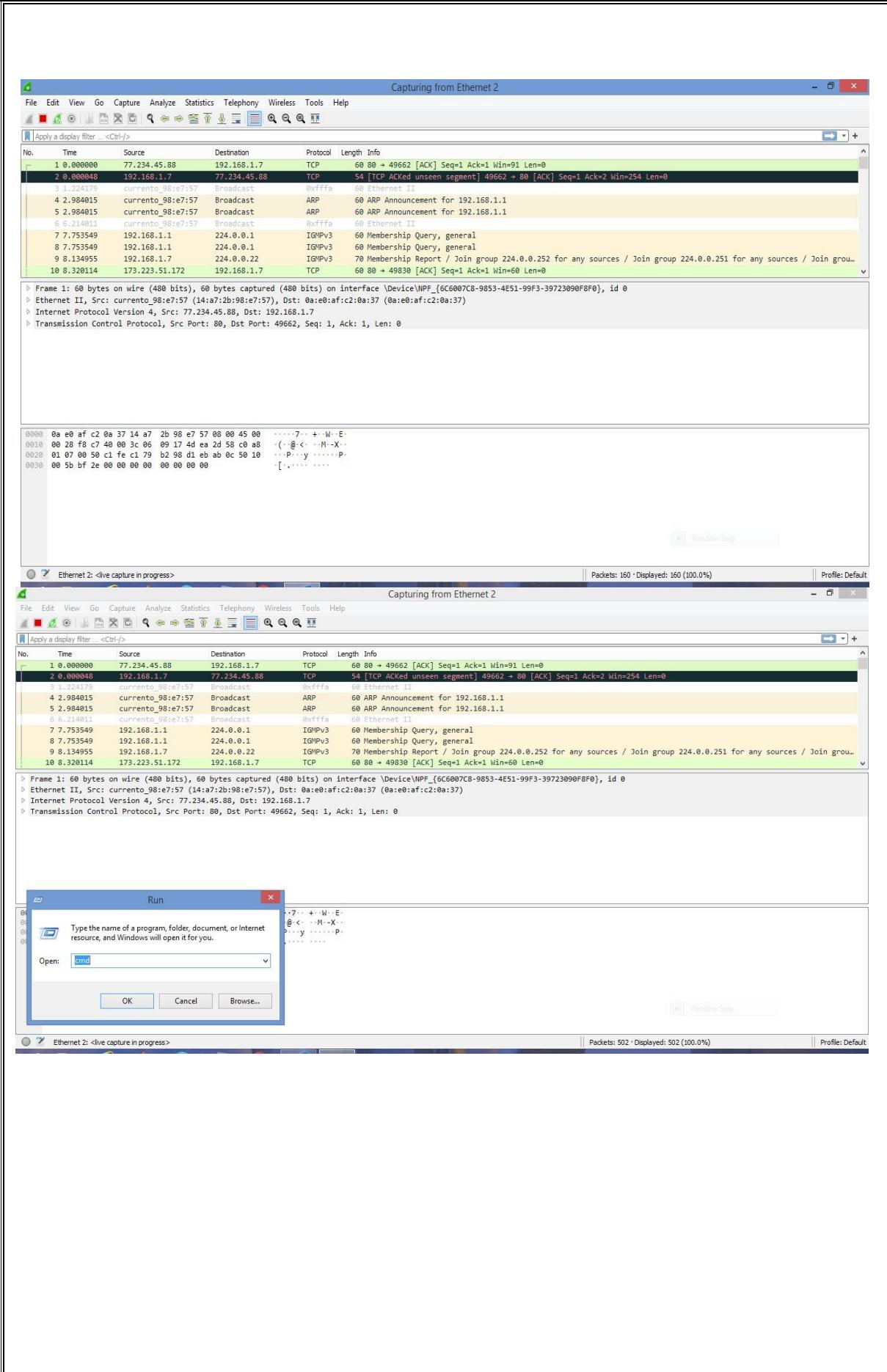


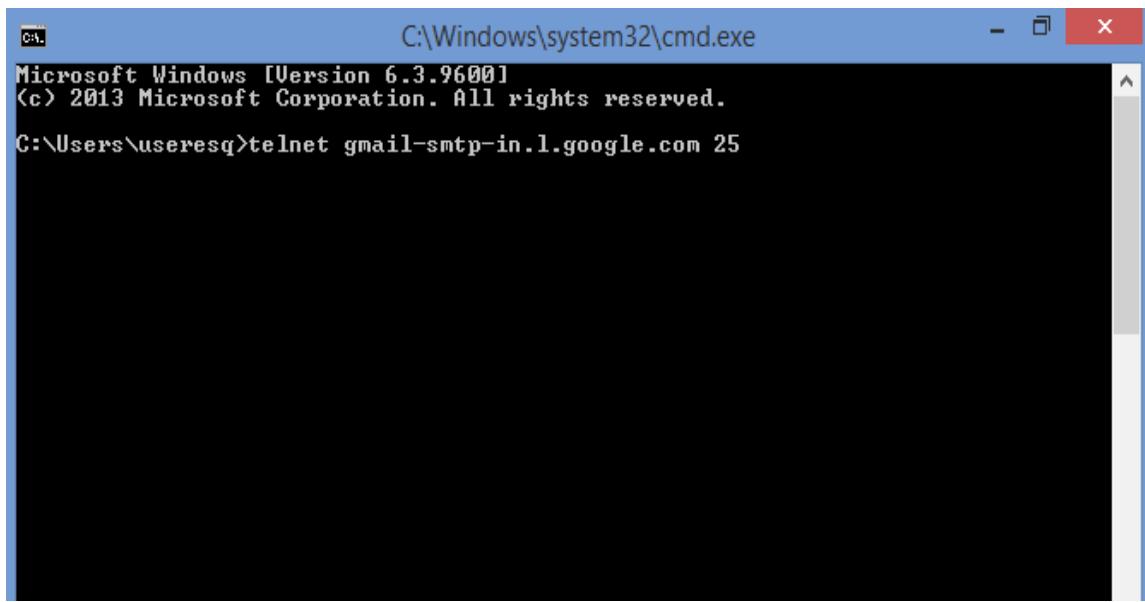


## Filtering SMTP Packets



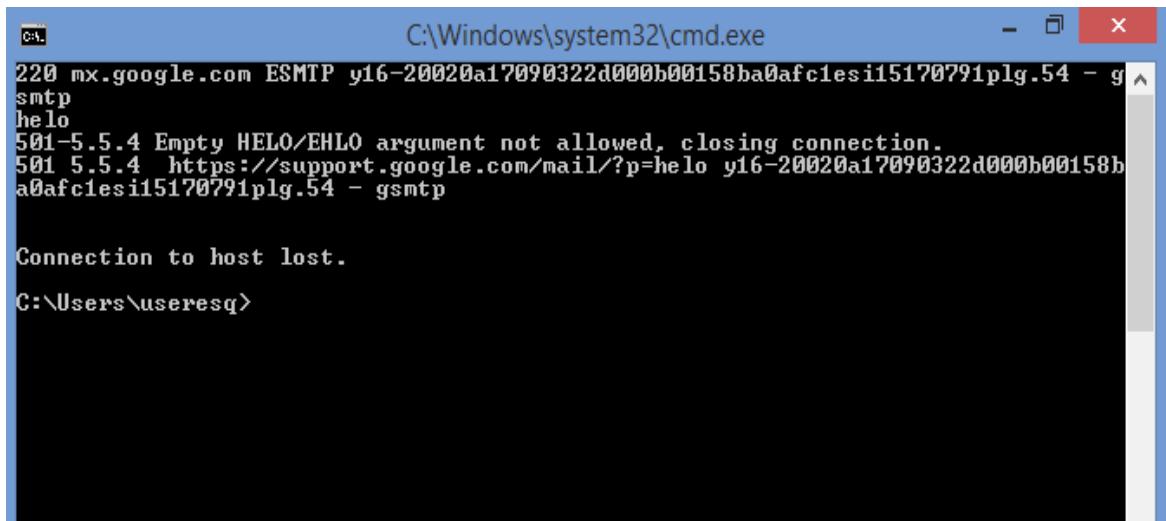
DEPARTMENT OF COMPUTER APPLICATIONS





```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\useresq>telnet gmail-smtp-in.l.google.com 25
```

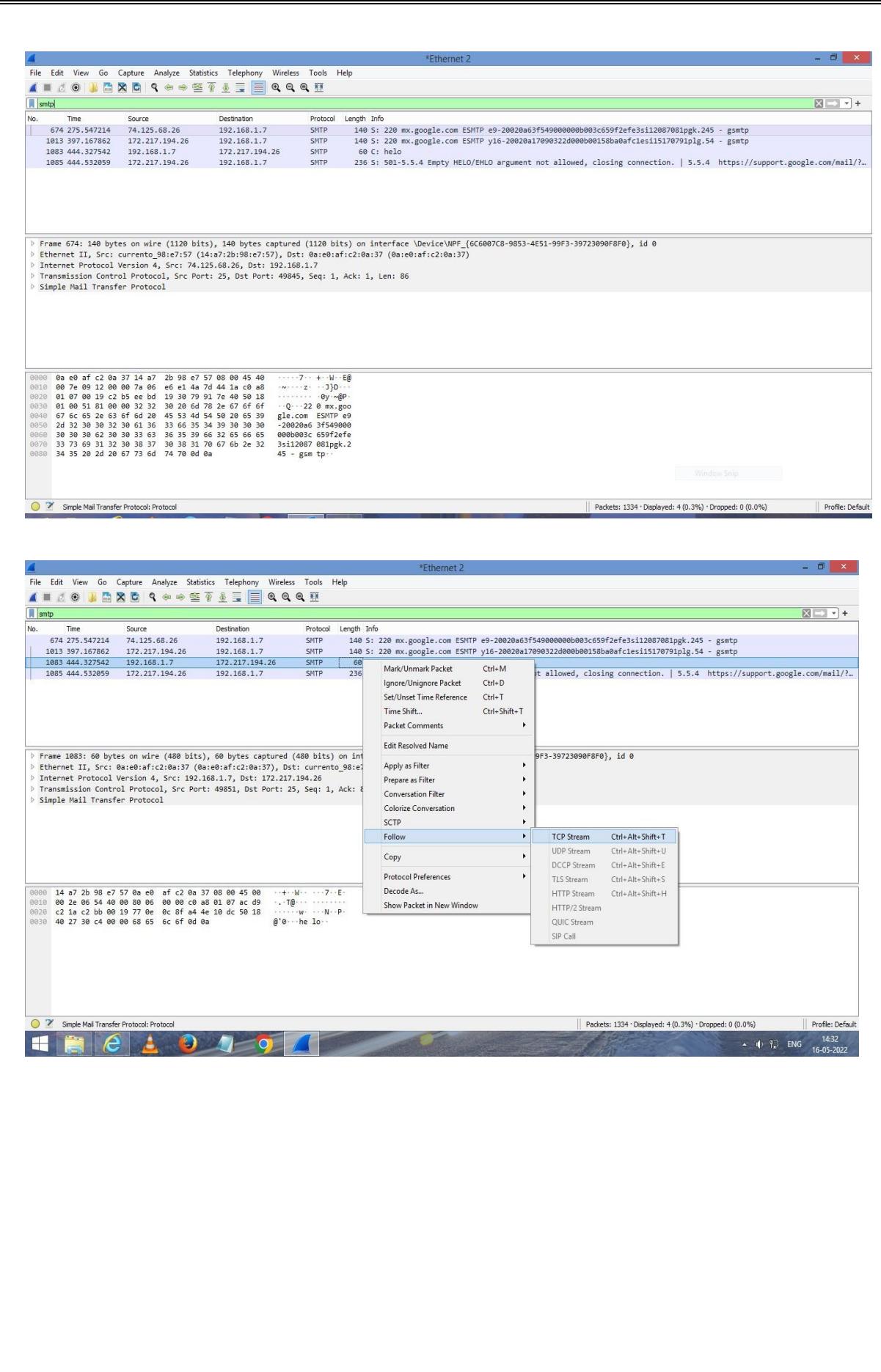


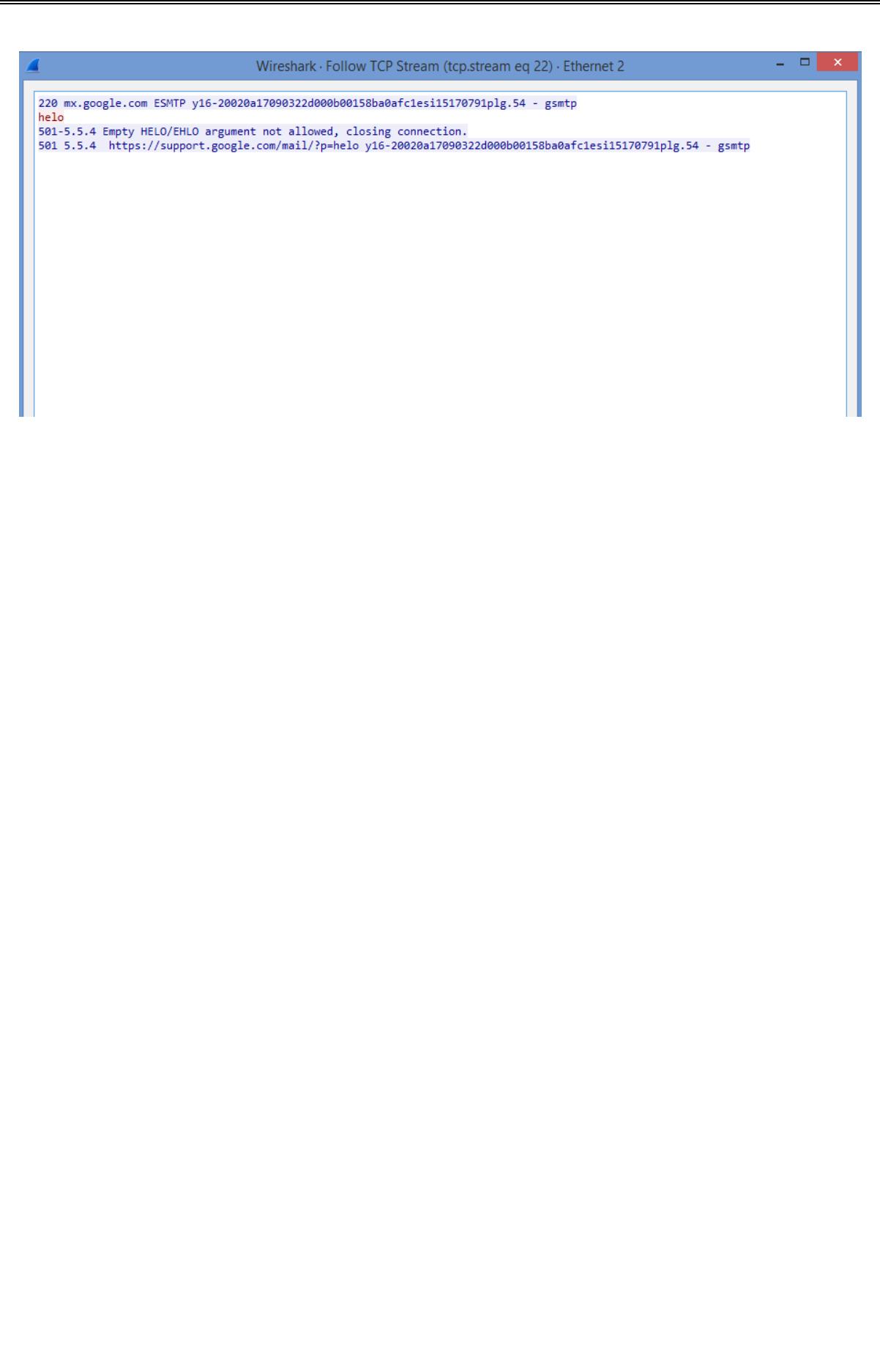
```
C:\Windows\system32\cmd.exe
220 mx.google.com ESMTP y16-20020a17090322d000b00158ba0afc1esi15170791plg.54 - g
smtp
helo
501-5.5.4 Empty HELO/EHLO argument not allowed, closing connection.
501 5.5.4 https://support.google.com/mail/?p=helo y16-20020a17090322d000b00158b
a0afc1esi15170791plg.54 - gsmtp

Connection to host lost.

C:\Users\useresq>
```

DEPARTMENT OF COMPUTER APPLICATIONS





## **EXPERIMENT-9**

### **LARAVEL SERVER**

#### **Step 1 – Install Apache Web Server**

Let's open up a Terminal and do first thing first update your package list using Sudo apt update command.

**\$ sudo apt update**

After updating your package list install apache webserver

**\$ sudo apt install apache2**

**\$ systemctl status apache2**

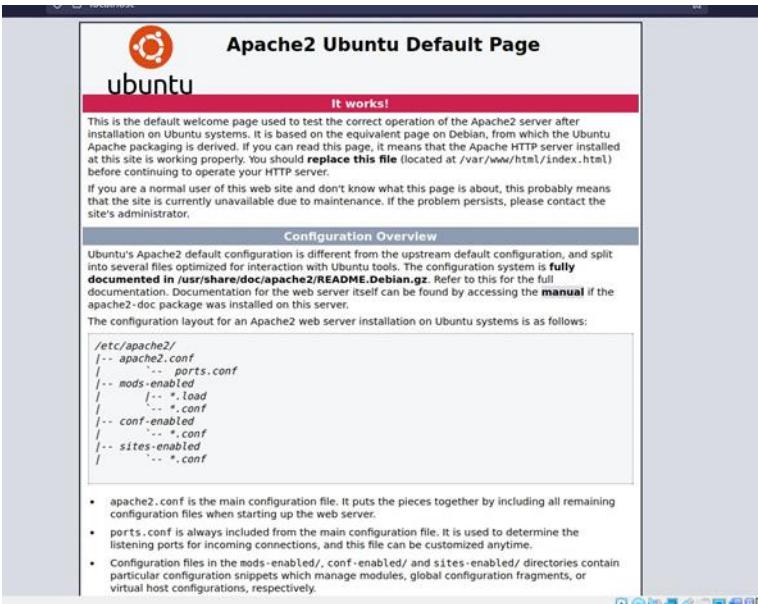
```
ebin@ebin-VirtualBox:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.11.0-25-generic linux-hwe-5.11-headers-5.11.0-25 linux-image-5.11.0-25-generic linux-modules-5.11.0
  linux-modules-extra-5.11.0-25-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following packages will be upgraded:
  apache2 apache2-bin apache2-data apache2-utils
4 to upgrade, 0 to newly install, 0 to remove and 78 not to upgrade.
Need to get 1,518 kB of archives.
After this operation, 4,096 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2 amd64 2.4.41-4ubuntu3.5 [95.5 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2-bin amd64 2.4.41-4ubuntu3.5 [1,180 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2-data all 2.4.41-4ubuntu3.5 [159 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2-utils amd64 2.4.41-4ubuntu3.5 [84.2 kB]
Fetched 1,518 kB in 1s (1,216 kB/s)
```

Now, check the status of apache server whether it is running or not. If the Apache server not running then use the following command to start apache serve and add to boot startup.

**\$ systemctl enable apache2**

```
ebin@ebin-VirtualBox:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-09-28 13:53:55 IST; 6min ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 2749 (apache2)
    Tasks: 7 (limit: 4650)
   Memory: 14.8M
      CGroup: /system.slice/apache2.service
              ├─2749 /usr/sbin/apache2 -k start
              ├─2750 /usr/sbin/apache2 -k start
              ├─2753 /usr/sbin/apache2 -k start
              ├─2754 /usr/sbin/apache2 -k start
              ├─2755 /usr/sbin/apache2 -k start
              ├─2756 /usr/sbin/apache2 -k start
              └─2757 /usr/sbin/apache2 -k start

Sep 28 13:53:55 ebin-VirtualBox systemd[1]: Starting The Apache HTTP Server...
Sep 28 13:53:55 ebin-VirtualBox apachectl[2747]: AH00558: apache2: Could not reliably determine the server's fully qualified name, using ebin-VirtualBox for Port 80
Sep 28 13:53:55 ebin-VirtualBox systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
```



To install Laravel 8.x, at least you must have PHP  $\geq$  7.3 on your system. And by default, the official Ubuntu 20.04 repository provides PHP 7.4 packages. Install PHP 7.4 packages using the apt command below.

```
$ sudo apt install libapache2-mod-php php php-common php-
xml php-gd php-opcache php-mbstring php-tokenizer php-json
php-bcmath php-zip unzip
```

```
ebin@ebin-VirtualBox:~$ sudo apt install libapache2-mod-php php php-common php-xml php-gd php-opcache php-mbstring php-tokenizer php-json
php-bcmath php-zip unzip
[sudo] password for ebin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package php-opcache is a virtual package provided by:
  php8.1-opcache 8.1.0-rc2-1+ubuntu20.04.1+deb.sury.org+1
  php8.0-opcache 8.0.11-1+ubuntu20.04.1+deb.sury.org+1
  php7.4-opcache 7.4.24-1+ubuntu20.04.1+deb.sury.org+1
  php7.3-opcache 7.3.31-1+ubuntu20.04.1+deb.sury.org+1
  php7.2-opcache 7.2.34-24+ubuntu20.04.1+deb.sury.org+1
  php7.1-opcache 7.1.33-41+ubuntu20.04.1+deb.sury.org+1
  php7.0-opcache 7.0.33-54+ubuntu20.04.1+deb.sury.org+1
  php5.6-opcache 5.6.40-54+ubuntu20.04.1+deb.sury.org+1
You should explicitly select one to install.
[sudo] password for ebin:
```

```

ebin@ebin-VirtualBox:~$ php7.4-dev php7.4-zip php7.4-mbstring php7.4-mysql php7.4-xml curl -y
php7.4-dev: command not found
ebin@ebin-VirtualBox:~$ sudo apt install php7.4 libapache2-mod-php7.4 php7.4-curl php-pear php7.4-gd php7.4-dev php7.4-zip php7.4-mbstring php7.4-mysql php7.4-xml curl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.11.0-25-generic linux-hwe-5.11-headers-5.11.0-25 linux-image-5.11.0-25-generic linux-modules-5.11.0-25-generic
  linux-modules-extra-5.11.0-25-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  autoconf automake autopoint autotools-dev debhelper dh-autoreconf dh-strip-nondeterminism dwz gettext intltool-debian
  libarchive-cpio-perl libarchive-zip-perl libcroco3 libdebservice-perl libfile-stripnondeterminism-perl libltdl-dev
  libmail-sendmail-perl libpcre2-16-0 libpcre2-32-0 libpcre2-8-0 libpcre2-dev libpcre2-posix2 libsigsegv2 libssl-dev
  libsub-override-perl libsys-hostname-long-perl libtinfo5 m4 php7.4-clip php7.4-common php7.4-json php7.4-opcache php7.4-readline
  pkg-php-tools po-debconf shtool
Suggested packages:
  autoconf-archive gnu-standards autoconf-doc dh-make gettext-doc libasprintf-dev libgettextpo-dev libtool-doc libssl-doc gfortran
  fortran95-compiler gcj-jdk m4-doc dh-php libmail-box-perl
The following NEW packages will be installed:
  autoconf automake autopoint autotools-dev curl debhelper dh-autoreconf dh-strip-nondeterminism dwz gettext intltool-debian
  libapache2-mod-php7.4 libarchive-cpio-perl libarchive-zip-perl libcroco3 libdebservice-perl libfile-stripnondeterminism-perl
  libltdl-dev libmail-sendmail-perl libpcre2-16-0 libpcre2-dev libpcre2-posix2 libsigsegv2 libssl-dev libsub-override-perl
  libsys-hostname-long-perl libtinfo5 m4 php-pear php7.4 php7.4-clip php7.4-common php7.4-curl php7.4-dev php7.4-gd php7.4-json
  php7.4-mbstring php7.4-mysql php7.4-opcache php7.4-readline php7.4-xml php7.4-zip pkg-php-tools po-debconf shtool
The following packages will be upgraded:
  libpcre2-32-0 libpcre2-8-0
2 to upgrade, 45 to newly install, 0 to remove and 76 not to upgrade.
Need to get 13.0 MB of archives.
After this operation, 54.0 MB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libsigsegv2 amd64 2.12-2 [13.9 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/main amd64 m4 amd64 1.4.18-4 [199 kB]
Get:3 http://ppa.launchpad.net/ondrej/php/ubuntu/focal/main amd64 libpcre2-8-0 amd64 10.36-2+ubuntu20.04.1+deb.sury.org+2 [201 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal/main amd64 autoconf all 2.69-11.1 [321 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal/main amd64 autotools-dev all 20180224.1 [39.6 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu focal/main amd64 automake all 1:1.16.1-4ubuntu6 [522 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu focal/main amd64 autopoint all 0.19.8.1-10build1 [412 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7.68.0-1ubuntu2.7 [161 kB]

```

Now go ahead and make tweak changes in PHP ini file and set cgi.fix\_pathinfo set to be 0. If this number is kept as a 1, the php interpreter will do its best to process the file that is as near to the requested file as possible. This is a possible security risk. If this number is set to 0, conversely, the interpreter will only process the exact file path—a much safer alternative.

**\$ cd /etc/php/7.4/apache2 \$ sudo nano php.ini**

Press **ctrl+w** and search for the word “cgi.fix” the uncomment the line and set it to 0.

...

**cgi.fix\_pathinfo=0**

```

ebin@ebin-VirtualBox:~$ cd /etc/php
ebin@ebin-VirtualBox:/etc/php$ ls
7.4 8.0
ebin@ebin-VirtualBox:/etc/php$ cd 7.4/
ebin@ebin-VirtualBox:/etc/php/7.4$ ls
apache2 cli mods-available
ebin@ebin-VirtualBox:/etc/php/7.4$ cd apache2/
ebin@ebin-VirtualBox:/etc/php/7.4/apache2$ ls
conf.d php.ini
ebin@ebin-VirtualBox:/etc/php/7.4/apache2$ sudo nano php.ini
ebin@ebin-VirtualBox:/etc/php/7.4/apache2$ sudo nano php.ini
ebin@ebin-VirtualBox:/etc/php/7.4/apache2$ █

```

```
GNU nano 4.8                               php.ini
; **You CAN safely turn this off for IIS, in fact, you MUST.**
; http://php.net/cgi.force-redirect
:cgi.force_redirect = 1

; if cgi.nph is enabled it will force cgi to always sent Status: 200 with
; every request. PHP's default behavior is to disable this feature.
:cgi.nph = 1

; if cgi.force_redirect is turned on, and you are not running under Apache or Netscape
; (iPlanet) web servers, you MAY need to set an environment variable name that PHP
; will look for to know it is OK to continue execution. Setting this variable MAY
; cause security issues, KNOW WHAT YOU ARE DOING FIRST.
; http://php.net/cgi.redirect-status-env
:cgi.redirect_status_env =

; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for CGI. PHP's
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not grok
; what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. Setting
; this to 1 will cause PHP CGI to fix its paths to conform to the spec. A setting
; of zero causes PHP to behave as before. Default is 1. You should fix your scripts
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; http://php.net/cgi.fix-pathinfo
:cgi.fix_pathinfo=1

; if cgi.discard_path is enabled, the PHP CGI binary can safely be placed outside
; of the web tree and people will not be able to circumvent .htaccess security.
:cgi.discard_path=1

; FastCGI under IIS supports the ability to impersonate
; security tokens of the calling client. This allows IIS to define the
; security context that the request runs under. mod_fastcgi under Apache

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify     ^C Cur Pos     M-U Undo
^X Exit          ^R Read File    ^L Replace      ^U Paste Text   ^T To Spell    ^A Go To Line  M-E Redo
```

```
GNU nano 4.8                               php.ini
; **You CAN safely turn this off for IIS, in fact, you MUST.**
; Thunderbird Mail/cgi.force-redirect
:cgi.force_redirect = 1

; if cgi.nph is enabled it will force cgi to always sent Status: 200 with
; every request. PHP's default behavior is to disable this feature.
:cgi.nph = 1

; if cgi.force_redirect is turned on, and you are not running under Apache or Netscape
; (iPlanet) web servers, you MAY need to set an environment variable name that PHP
; will look for to know it is OK to continue execution. Setting this variable MAY
; cause security issues, KNOW WHAT YOU ARE DOING FIRST.
; http://php.net/cgi.redirect-status-env
:cgi.redirect_status_env =

; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for CGI. PHP's
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not grok
; what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. Setting
; this to 1 will cause PHP CGI to fix its paths to conform to the spec. A setting
; of zero causes PHP to behave as before. Default is 1. You should fix your scripts
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; http://php.net/cgi.fix-pathinfo
:cgi.fix_pathinfo=0

; if cgi.discard_path is enabled, the PHP CGI binary can safely be placed outside
; of the web tree and people will not be able to circumvent .htaccess security.
:cgi.discard_path=1

; FastCGI under IIS supports the ability to impersonate
; security tokens of the calling client. This allows IIS to define the
; security context that the request runs under. mod_fastcgi under Apache

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify     ^C Cur Pos     M-U Undo
^X Exit          ^R Read File    ^L Replace      ^U Paste Text   ^T To Spell    ^A Go To Line  M-E Redo
```

Press Ctrl + x then y to Save and Exit.

Now Restart The apache service.

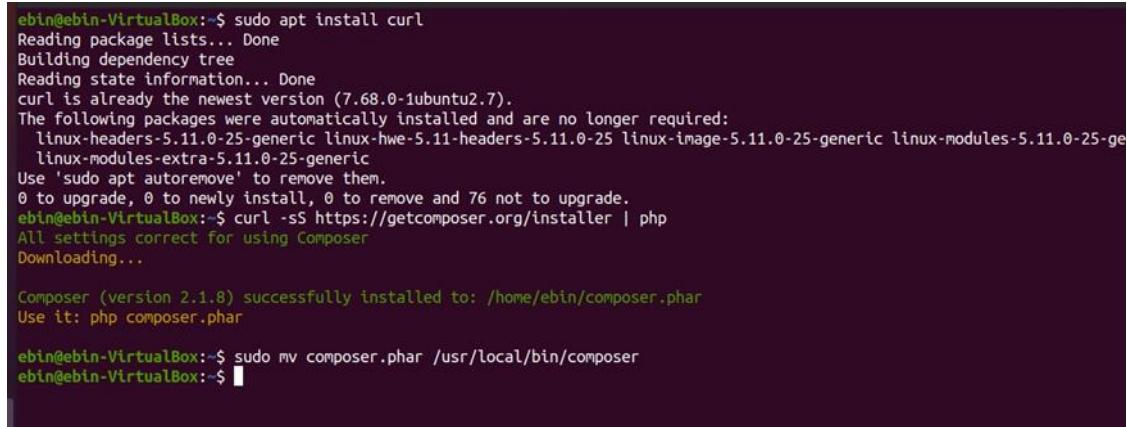
**\$ systemctl restart apache2**

### **Step 3 – Install Composer PHP Packages Management**

Install the composer package manager go ahead and download and install

Composer. and move the composer .phar file to  
usr/local/bin/composer directory.

```
$ sudo apt install curl $ curl -sS https://getcomposer.org/installer | php $ sudo
mv composer.phar /usr/local/bin/composer
```



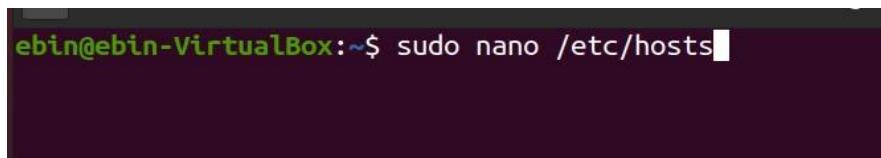
```
ebin@ebin-VirtualBox:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.68.0-1ubuntu2.7).
The following packages were automatically installed and are no longer required:
  linux-headers-5.11.0-25-generic linux-hwe-5.11-headers-5.11.0-25 linux-image-5.11.0-25-generic linux-modules-5.11.0-25-ge
  linux-modules-extra-5.11.0-25-generic
Use 'sudo apt autoremove' to remove them.
0 to upgrade, 0 to newly install, 0 to remove and 76 not to upgrade.
ebin@ebin-VirtualBox:~$ curl -sS https://getcomposer.org/installer | php
All settings correct for using Composer
Downloading...

Composer (version 2.1.8) successfully installed to: /home/ebin/composer.phar
Use it: php composer.phar
ebin@ebin-VirtualBox:~$ sudo mv composer.phar /usr/local/bin/composer
ebin@ebin-VirtualBox:~$
```

As you are in a local environment you need a local dns resolver for your site. Go ahead and edit /etc/hosts file, add a dns record for your site then save the file.

**\$ sudo nano /etc/hosts**

**127.0.0.1 myapp1.com**



```
ebin@ebin-VirtualBox:~$ sudo nano /etc/hosts
```

```
GNU nano 4.8                               /etc/hosts
127.0.0.1      localhost
127.0.1.1      ebin-VirtualBox

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

127.0.0.1    myapp1.com

[ Read 11 lines ]
^G Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify
^X Exit       ^R Read File   ^A Replace   ^U Paste Text  ^T To Spell
```

Now get back to the web browser and open a tab then type your project host name.

